Reliable Operational Voltage Minimization for Nanometer SRAMs

A Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

In partial fulfillment

of the requirements for the degree of

Doctor of Philosophyin Electrical Engineering

by

Jiajing Wang

August 2009

Approval Sheet

The dissertation is submitted in partial fulfilment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering

Jiajing Wang (AUTHOR)

This dissertation has been read and approved by the examining committee

Dr. Benton Calhoun (Advisor)

Dr. Mircea Stan (Committee chair)

Dr. John Lach (Committee member)

Dr. Joanne Dugan (Committee member)

Dr. Kevin Skadron (Committee member)

Accepted for the School of Engineering and Applied Science:

Dean, School of Engineering
and Applied Science

(August, 2009)

Research advisor
Benton H. Calhoun

Author
Jiajing Wang

Reliable Operational Voltage Minimization

for Nanometer SRAMs

Abstract

Since Static Random Access Memory (SRAM) continues to be the largest component in many embedded digital systems or System-on-Chips (SoCs), its power consumption dominates the overall power of the system. To reduce power, a low voltage SRAM is highly demanded. SRAM minimum operational voltage (Vmin) is determined by the lowest acceptable functional yield in the presence of variation. In this thesis, we propose new design methods to combat both local and global variation and achieve maximum power savings while maintaining acceptable yield.

Local variation causes a distribution of Vmin for cells on the same array. The tail value of the distribution determines the minimum V_{DD} for the whole array. For large SRAMs, this tail event occurs once out of millions of samples. For such rare events, standard Monte Carlo method becomes intolerable expensive. Recently, some fast Monte Carlo and non-Monte Carlo methods have been proposed recently for rare event estimation and applied successfully to SRAM circuits. However, they require either complicated computations or

special care of sampling. In this thesis, we propose a fast and accurate method dedicated for SRAM Vmin and yield estimation. It requires only a small scale of Monte Carlo simulations on static noise margin and then applies dedicated statistical models for Vmin and yield. This method is generic for estimating Vmin due to hold, read, and write failures. The model offers $> 10^4$ speed-up and $<$1% error relative to standard Monte Carlo. It also matches well with other fast Monte Carlo methods for tails up to 8σ, but shows less complexity and smaller estimation variance.

SRAM Vmin also shifts with global variation. The conventional worst case approach limits power savings for normal conditions. Instead, adaptive method can track and compensate global variation and thus allow maximum power savings. Thus we propose an adaptive standby voltage scaling system based on a tunable canary replica cell. We present circuits for robustly building the control logic that implements the feedback mechanism at subthreshold supply voltages. Data reliability is ensured by a critical failure threshold, which can be programmed for trading off leakage power savings with yield. We also propose several techniques to enhance the adaptiveness and automation of the canary system. Silicon results from both 90nm and 45nm test chips demonstrate the function of the canary system. To realistically quantify the potential savings achievable by the canary scheme, we assess the impact of various sources of overhead. Finally, we investigate the performance of the canary based scheme in nanometer technologies, and we show that it promises to provide substantial standby power savings down to the 22nm node.

Acknowledgments

Blabla...

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

1.1.1 Motivation of Low Voltage SRAM

Static random accessed memory (SRAM) has been and continues to be the largest component in a chip. It is expected to occupy over 90% of the area of system-on-chip (SoC) by 2013 [47]. As a result, its power consumption becomes a big concern. Various applications demand SRAMs to operate at a lower voltage for power reduction.

SRAM is commonly used in high performance applications, such as high speed processors. For high performance applications, leakage power becomes a big concern because leakage current grows dramatically as technology scales. SRAM leakage power often dominates the total leakage power of the chip due to its large area. Therefore, it is highly demanded to reduce SRAM standby power. Leakage current consists of subthreshold leakage current, gate leakage current, drain induced leakage current and junction current. Several

techniques have been proposed to reduce SRAM leakage current. Dual-V_T [73] and body biasing [34] utilized V_T control to reduce subthreshold leakage current. Source biasing [1, 76] and voltage scaling [37] approaches collapse the actual rail-to-rail voltage of SRAM cells so that both the subthreshold and gate leakage current can be reduced. Because of the advantage of gate leakage reduction, voltage scaling and source biasing become more effective for leakage power reduction in deeply scaled technologies. Therefore, an SRAM with a lowered standby voltage is desired.

In recent years, the demand of the low-power SoCs for portable electronics grows rapidly. To extend the battery life of the portable devices, dynamic voltage and frequency scaling (DVFS) is widely used to reduce active power and energy by adapting voltage and frequency of the system when the workload decreases. To implement a DVFS system, an SRAM with a wide-range of lower operational voltage is an indispensable component. Great efforts have been made to push down SRAM operational voltage within SoC system to improve power savings (e.g. [3, 50, 67]). For ultra-low power and energy constrained applications like implantable medical devices and wireless sensor networks, the operational voltage for minimum energy often occurs near or below threshold voltage (V_T). This requires an SRAM also operate at such low voltage ([8, 11, 36, 61, 75]).

Besides power reduction, another benefit from lower operational voltage is the suppression of some aging related reliability issues, such as negative bias temperature instability (NBTI) and hot carrier injection (cite a paper). They have weaker effects on SRAM under lower supply voltage.

1.1.2 Challenges of Low Voltage SRAM

Although lowered supply voltage can effectively reduce power consumption and mitigate some aging reliability issue, it degrades SRAM functionality and thus functional yield, especially in the presence of variations.

SRAM Functional Yield

There are four failure mechanisms related to SRAM functional yield: hold failure, read failure, write failure, and access failure. Hold failure occurs when the cell does not have adequate noise margin to preserve data in the occurrence of soft error noise sources like radiations and alpha particles. Read failure is caused by the disturbance on the node holding '0' upon read and results in the flipping of data after read. Write failure is mainly caused by the incapability of pulling down the node initially holding '1' and thus the cell data is unable to toggle within the required write time. These three categories of failures are mainly determined by SRAM cell stability. The access failure occurs when an insufficient voltage difference on bitlines is developed for read sensing within the required timing period. It is impacted not only by the ratio of cell on current versus BL leakage current but also by the performance of the sensing circuit (e.g. offset of the sense amplifier). When SRAM operates actively, yield is determined by all of the failure types. On the other hand, when SRAM operates inactively (i.e. at standby mode), the hold failure determines the yield.

Typically SRAM is designed to have sufficient hold, read, and write noise margins as well as access speed under the nominal supply voltage. However, lowered operational voltage degrades each of the functionalities and results in a lower yield and reliability. Therefore, there exists a trade-off between power and yield with regard to operational voltage. A

critical figure of metric direct related to this trade-off is the minimum operational voltage (Vmin) for maintaining a lowest acceptable yield and determining the maximum achievable power reduction. Vmin becomes one of the biggest concerns for SRAM design. Using an under-estimated Vmin causes intolerable failures and decrease SRAM yield. On the other hand, using an over-estimated Vmin wastes power and energy. Vmin is dependent on the sensitivity of each functional failure to operational voltage. In old technologies, like other circuit metrics, Vmin is a deterministic value. However, technology scaling beyond 90nm has posed increased parameter variations, which change all the circuit problems including Vmin from deterministic to stochastic.

Variation

For more than three decades, technology scaling has been contributing to the significant improvement of density and performance for IC designs. However, continuous technology scaling in sub-100nm region is posing serious challenges to circuit designs. One of the biggest challenges is increased variation.

Based on the scale of variation, all the variations can be categorized into two groups: (1) *global* variation and (2) *local* variation. Global variations occur on the die-to-die scale and influence all the transistors on the same die. They mainly include the inter-die manufacture related *process* variations and environmental conditions including *voltage* supply fluctuation and *temperature* change, i.e. PVT variations [7]. On the contrary, local variations occur within die. They have different effects on individual transistors and thus cause mismatch between adjacent ones. Local variations also include process, voltage, and temperature variations within chip. Local voltage and temperature variations are mainly caused

by uneven power dissipation across the die due to variations in switching activity. Local process variations can be further classified as *systematic* and *random* variation based on the nature of variation. Systematic variation such as layout dependent variation is predictable and can be modeled as a function of deterministic factors such as layout structure and the surrounding topological environment [70]. Since typically, the layout of SRAM cell is carefully designed and the layout pattern within the whole SRAM array is quite regular, we assume layout dependent systematic variation is very small. On the contrary, random variation is unpredictable and must be modeled as stochastic events. Random Dopant Fluctuation (RDF) and line edge roughness are the major sources of V_T random variation, and RDF becomes the dominant one as device dimension continues shrinking.

SRAM is extremely suspectible to local variations because of three reasons. First, SRAM cell commonly uses transistors with the smaller geometry for higher density. RDF induced V_T random variation is normally distributed with the standard deviation (σ) inversely proportional to the channel area [2]. As a result, SRAM cell transistors have random V_T variation with a larger value of σ. Second, SRAM contains a huge number of identical cells. Modern SRAMs often contain millions of cells. With continuous shrinking of device dimensions and growing demand of larger capacity of embedded memory on chip, billions or trillions of SRAM cells can be integrated into one chip. With such a big number of samples, large variations beyond 5σ/6σ are likely to happen. Third, many SRAM metrics are very sensitive to mismatch because SRAM cell commonly uses two symmetrical inverters cross-coupled connected. A small mismatch between adjacent transistors within the two inverters can lead to a large variation in the cell's behavior, such as cell stability.

The impacts of variation on SRAM Vmin are twofold. First, random variation spreads

the Vmin of cells on the same SRAM array under the same operating condition. The minimum voltage for the whole SRAM is determined by the worst value at the tail of the distribution. Because of the nature of randomness, the tail value in one die might be different with that in another die. Second, global variation primarily shifts the mean of the distribution and adds more uncertainty of SRAM Vmin.

1.1.3 Existing Solutions

As the limitation of CMOS technology, new technologies for silicon devices such as multiple gate and new devices beyond silicon such as carbon nanotube and organic molecular transistors are emerging (cite some). However, these new technoglies and devices will not be mature in the near future. CMOS technology will continue scaling and thus solutions to overcome limitations from scaling are extremely important for SRAM design beyond 45nm.

Functional Yield Improvement

Because of the combination of density, performance, and compatibility with CMOS logic process, conventional 6T SRAM continues to play a dominant role in deeply scaled technologies. However, it is encountering greater difficulties to maintain sufficient functional margins in lower operating voltage as variation increases in advanced technologies. In recent years, various read and write assist methods have been proposed to improve the functionality of 6T SRAM cell and push down its Vmin for more power reduction. The key of all the assist methods is to alter the strength of the latch inverters and/or the amplitude/duration of the noise source to favor read or write operation [54]. They either change

the voltage of different terminals on the cell(e.g. [3, 31, 44, 49, 56, 71, 77]) or change the width of the pulse on the terminal (e.g. [33, 50]).

Instead of remedying the conventional 6T cell, an alternative attempt is to use novel SRAM cells. Adding a read buffer in the cell can eliminate the limit of read stability. This kind of cells include 8T cell [12, 13] and different versions of 10T cell [8, 11, 36]. [38] also proposes a 10T cell with new storage latch based on Schmitt Trigger inverter. These novel cells have been reported to function at a lower voltage than 6T cell at the same condition.

Such a variety of circuit options makes it possible to implement low voltage SRAM in spite of large variations. Meanwhile, the complexity for searching the optimum solution grows. To pick the most effective method for a given technology and application, we need evaluate all the possible methods. For each method, we have to try different values of the tuning variables and evaluate the yield improvement for each. The whole analysis process can be intolerable slow if the individual run takes a long time. Thus a fast and accurate method for yield estimation is highly desired. With a fast design method, we can quickly eliminate less effective circuit options in the early design phase, and then perform more thorough and accurate analysis on a smaller number of candidates and finally find the optimum one.

Statistical Design

The traditional circuit design method uses the worst case based analysis. Designers simulate circuit with a worst-case corner for PVT variation and aging This ensures the circuit function correctly under the worst operating condition. However, this deterministic design method results in over-design for normal conditions. In the terms of Vmin, this

can cause a significant waste of power and energy. Because of the stochastic nature of random variations, statistical design method should be used instead [26]. However, since SRAM often contains millions of cells, an acceptable yield requires an extreme small failure probability, which is in the order of magnitude of 1 part-per-million. For such a rare event, Monte Carlo, the standard statistical simulation method, becomes prohibitively slow. To speed up the estimation of the rare events, a variety of methods arise and fall into two major categories: non-Monte-Carlo (non-MC) methods and improved Monte-Carlo (MC) methods.

Non-MC methods include comprehensive analytical models for SRAM metric and generic mathematic methods like the boundary searching approach [24]. However, they are often intractable when the circuit contains a large number of random variables. For such a circuit, MC simulation is more straightforward. Thus methods to improve MC becomes attractive. One way to reduce MC run time is to hasten the generation of the rare events. Interesting techniques include Importance Sampling (IS) [30] and the Statistical Blockade (SB) tool [57]. However, the efficiency of IS and the SB tool relies on the goodness of the sampling distribution and the tail filter respectively. Thereby other fast methods with comparable accuracy but simple procedure can be appealing.

Adaptive Design

To tolerant variation, the traditional solution is to add guard-band margin for worst case variations and aging conditions. Although it is very robust, substantial power and energy are wasted. A better solution to deal with variation uncertainty is the adaptive design, which dynamically adjusts circuit operation to runtime workload, environmental

and process variations. It usually consists of two phases: (1) PVT variation detection and (2) compensation/correction. Various adaptive methods with different detection methods and/or compensation methods have been proposed for SRAM.

The first group of adaptive methods uses on-chip process skew sensors and thermal sensors to detect PT variations and then adjust assist knobs (body bias, VDD, VSS, VWL etc) to compensate the impact of variation. For instance, authors in [46, 71] use on-chip SRAM leakage monitor to detect global V_T variation, and then adjust body bias to compensate V_T variation for SRAM yield improvement. The similar method is also used for microprocessor performance and leakage trade-off [60]. Ring-oscillator delay monitor is also proposed to detect V_T variation [45]. [31] proposes to use the thermal/process sensor to track PT variation and aging and then tune the assist knob VWL for yield improvement and Vmin reduction. However, this kind of methods requires a conversion from the amount of the measured variation to the amount of the required assist bias value. This makes the adaptiveness less effective.

Another way to adapt process variation is to perform post-silicon calibration during initial test. [74] directly measures SRAM performance, leakage and stability, and tune assist knobs to meet the target value. The required adjust values are saved on chip and loaded upon operation. Although this method can effectively compensate die-to-die and within-die process variation, it is passive and still needs margins for worst environmental and aging condition.

A more effective approach that can potentially eliminate all the margins for PVT variations is in-situ error detection. [6] proposes a new register that can detect timing and soft errors in-situ and then adjust frequency or supply voltage to reduce failures. Although

this kind of in-situ error detection is very attractive, it is impractical to use for SRAM cell because of its large area overhead and SRAM cell's stringent density requirement.

Another effective adaptive approach is canary circuit, which emulates the critical part of the actual design. Canary circuit that can mimic critical path delay in microprocessor has been reported to effectively track PVT variations and aging effects in [17]. [9] first proposes canary flip-flop that mimics the stability of the worst flip-flop in design to achieve power savings near the optimum. To track local variation, multiple canary circuits are used by different transistor sizing (e.g. the ratio of pmos/nmos is changed to implement different category of canary flip-flop [9]). However, sizing knob is less effective as parameter variation increases. In addition, the distribution of SRAM cell metric can be quite wide because of the big number of instances. To mimic the worst cell, the canary cell must be able to cover a wide range beyond 6σ. Therefore, novel canary circuit for SRAM is desired.

1.2 Major Contributions

In this thesis, we mainly address the impact of variation on SRAM minimum operational voltage in deeply scaled technologies. Design methods to combat both local random variation and global variation are presented to achieve the optimum power reduction while maintaining required yield.

A Fast and Accurate Method for Vmin and Yield Estimation

Within-die variation causes a skewed distribution of Vmin for individual cells in a memory array. Cells far out in the tail (i.e. $>6\sigma$) limit Vmin for large SRAMs and determine the maximum power reduction. A quick and accurate estimation of Vmin in the pres-

ence of random variations can accelerate the exploration of the potential trade-offs among performance, power, and yield and contribute to design an optimum SRAM for a given application. However, standard Monte Carlo simulation is too computationally expensive to estimate the tail values for large SRAMs. Comprehensive analytical model and boundary searching method are intractable when the number of random parameters is large. Fast Monte Carlo methods like the Statistical Blockade tool (SB) and Importance Sampling (IS) can hasten the generation of the rare events. However, their efficiency is sensitive to the goodness of the sampling distribution or the tail filter.

In this thesis, we propose a new fast and accurate method to estimate Vmin based on the statistical trend of static noise margin (SNM) with V_{DD} scaling. Statistical models for Vmin and yield are presented. We first show its efficiency for standby Vmin estimation. We then extend this method to estimate Vmin and cell yield for read and write operations, which are critical for dynamic power reduction. We generalize the model for both symmetrical and asymmetrical types of cells so that it can be used for any types of SRAM cells. Our method matches well with Monte Carlo within 5σ. In addition, it shows excellent agreements with other fast Monte Carlo methods (the SB tool for standby Vmin estimation and IS for active Vmin estimation) beyond 5σ, where it is extreme costly for Monte Carlo. With comparable accuracy, our method offers a speedup of more than 4 orders of magnitude over Monte Carlo, and it is less complicated than SB and IS. The analysis of the statistical accuracy shows that estimates from our method maintain less than 5% error with 95% confidence out to 8σ.

An adaptive standby V_{DD} scaling system for aggressive power reduction

Although designing for the worst-case ensures stability in all the conditions, it over-protects data for non-worst-cases and hence loses extra power savings. An adaptive design that can compensate for variation is preferable for ultra-low-power applications. Although active feedback with opamp and programmable reference voltage [25, 32] can track both process variation and voltage fluctuation, extra thermal sensor is still needed to track temperature change. [59] proposes to use replica circuits to track PVT variation. However, the replica cell it used has a much higher failure voltage than the actual SRAM cell.

In this thesis, we propose a closed-loop standby V_{DD} scaling system using canary replica cells. It can track all the PVT changes and achieve the maximum power savings while maintaining data retention. In addition, our method provides the flexibility to trade-off between the safety of data and decreased leakage power. A novel canary cell circuit is presented. We thoroughly analyze the adaptiveness of the canary cells for tracking PVT variations. We present circuits for robustly building the control logic that implements the feedback mechanism at subthreshold supply voltages. A prototype implemented on a 90nm test chip confirms the function of the canary cell and the closed-loop feedback circuits.

We also propose several techniques to improve the efficiency of this approach for 45nm and beyond technologies. We add dummy cells around canary cell to enhance the correct tracking of the layout dependent variations. We also propose a new canary circuit to avoid the possibility that the canary cell would never fail because it happens to hold its more-stable value. We incorporate a built-in self-test (BIST) block to automate the calibration of SRAM Vmin and the tuning of the initial failure threshold for adapting process variation. A 45nm test chip further demonstrates the effectiveness of the canary system for SRAM

standby power reduction in sub-45nm nodes. The canary scheme also shows substantial standby power savings for predictive technology nodes down to 22nm. We use conventional 6T SRAM as an example in our work, but the canary system can be extended for other SRAMs using different type of cell.

1.3 Organization

The remainder of this dissertation is organized as follows.

Chapter 2 describes the impact of random variation on data retention voltage, i.e. standby Vmin, and then presents the new statistical method for Vmin and yield estimation during standby operation. The distribution of SRAM hold SNM and its sensitivity to V_{DD} is investigated. Based on that, we develop the analytical model for standby Vmin. Finally, we show experiment results with 6T SRAM in a 90nm node and compare different methods in terms of accuracy, speed, and confidence.

In Chapter 3, we extend our statistical method for active Vmin estimation. We analyze read and write noise margin and their sensitivity to V_{DD}. Then we generalize the model for hold, read, and write. We show experiment results with 6T SRAM in a commercial 45nm node.

In Chapter 4, we address the impact of global variation on SRAM Vmin and then propose the canary based closed-loop standby V_{DD} scaling system. We first describe the principle of the closed-loop V_{DD} scaling system. Then we describe the details of the major components, including the canary cell and bank, the failure detector and the controller. Adaptive reaction and overhead sources are analyzed. Finally, we present the measurement results from the 90nm test chip.

Chapter 5 presents the enhanced canary system for more advanced technologies. We first describe the improvements on the canary cell. Then we present the built-in-self-test and tuning (BISTaT) block for self-calibration of the canary system. The implementation and measurement of the 45nm test chip with the new canary cell are presented. In addition, we show the simulation results with the predictive technology model down to 22nm.

Chapter 6 concludes this dissertation and points out possible future improvements and applications for the statistical method as well as the canary adaptive scheme.

Chapter 2

A Statistical Method for DRV Estimation

2.1 Motivation

Standby leakage power can dominate the total power budget of memories or System-on-Chips (SoCs) that dedicate increasingly large percentages of die area to memory. Supply voltage (V_{DD}) scaling is an effective approach for leakage power savings during SRAM/Cache standby mode. Besides the direct effect of smaller voltage on power saving, V_{DD} scaling reduces both sub-threshold leakage current due to drain induced barrier lowering (DIBL) effect and gate leakage current. Lowering V_{DD} as far as possible maximizes leakage power savings but might also lead to data loss. The data retention voltage (DRV) is the lower bound of the standby supply voltage that still preserves data in the bitcells.

Device variability has been a big challenge for circuit design in nanometer technologies. The most problematic variation is caused by the random inter-device variation sources, like

random doping fluctuation (RDF). RDF induced variation increases with technology scaling. The randomness of threshold voltage (VT) due to RDF can be modeled as a normal distribution with the standard deviation inversely proportional to the channel area. SRAM cells often use the smallest geometry to increase memory density, thus becoming particularly susceptible to RDF. Consequently, the DRV of one cell can be very different from another cell in the same array. Note that the DRV of an entire array is the DRV of the worst cell in the array. The random nature of the DRV of bitcells makes the array DRV also a random variable. Say, for a 1-Mb array, we desire that at least 99% of manufactured arrays have a DRV of 0.7 V or lower. This places a very strict yield requirement on the bitcell: the probability of the bitcell DRV exceeding 0.7 V must be 1.005e-8 (1 part per 100M) or less. For larger array sizes, this exceedance probability for the bitcell must be even lower. These extreme yield requirements on the bitcell DRV pose a difficult problem for yield estimation, given a bitcell design.

A straightforward method for obtaining the array DRV is to run a full Monte Carlo (MC) simulation until we obtain DRV values at the required probability levels. However, this is often prohibitively slow for multi-Mb memories (e.g. months on a single machine). For instance, to estimate the DRV with the probability of 1.005e-8, standard Monte Carlo should run at least 100 million sample points to reach such an extreme probability level. Even then, the estimate of DRV quantile will be suspect because of the lack of statistical confidence. However, running the requisite billions of samples (circuit simulations) is utterly intractable.

To speed up the estimation of the rare events, various methods arise and fall into the following two major categories.

1. **Non-Monte-Carlo (non-MC) methods**: The first non-MC method is to develop a dedicated comprehensive analytical model for the figure of metric. Although [51] propose a theoretical model to approximate the DRV of a single cell, they didnt address the statistical characteristics of DRV. The question of how variations impact the long tail of the DRV distribution is not answered. The second and more generic non-MC method is the boundary searching approach, which intends to find the boundaries in the parameter space that correspond to success/failure of the circuit without using MC sampling [24, 58]. The authors demonstrated its efficiency for SRAM read access yield estimation when considering only two major design parameters. However, the real access yield is also determined by other design parameters that have a minor impact on read access. When all the parameters are searched, this method becomes expensive too.

2. **Improved Monte-Carlo (MC) methods**: The huge expense of MC for rare event estimation is mainly due to the inefficiency of the rare event sampling. Importance sampling [15, 16, 30] and the Statistical Blockade (SB) tool [57] are two interesting techniques to hasten the generation of the rare events. However, their efficiency highly relies on the goodness of the sampling distribution and the tail filter respectively. Extrapolation is an alternative way to avoid a full MC simulation. We can run a relatively small number of samples and fit them into a known distribution. Then we can quickly acquire the estimates in the extreme tail region by simply calculating with the fitting distribution. Although it is very simple, the accuracy of the extrapolation method is dependent on how good the fitting distribution is. For the non-Gaussion variables like DRV, it is hard to find a proper known distribution that

can well fit the skewness of the tail region. Fitting as a normal and log-normal distribution either underestimate or overestimate the tail values. The SB tool proposes to use the generalized pareto distribution (GPD) to particularly fit the tail samples. Its accuracy is dependent on the number of tail samples, which also requires fast Monte Carlo methods like the tail filter in the SB tool to accelerate its generation.

In this chapter, we propose a new fast method to predict the tail of the DRV distribution. We use the extrapolation method so that only a small number of Monte Carlo samples is required. The high accuracy is achieved by using a dedicated statistical model for DRV. In the rest of this chapter, we first discuss the definition of DRV and the impact of local variation on DRV. Then we discuss SRAM hold static noise margin and its sensitivity to V_{DD}. Based on this, we develop the statisical model for SNM and DRV. Finally, we show experimental results and analyze the confidence of estimates.

2.2 DRV - Standby Vmin

Figure 2.1 shows the structure of the traditional 6T SRAM bitcell. NL and NR are pull-down transistors; PL and PR are pull-up transistors; and XL and XR are pass-gate transistors. During standby, the wordline signal WL is '0' and the two bitlines (BL and BLB) are precharged to V_{DD}. During standby/hold, the bitcell's job is to hold its data. The SRAM cell consumes leakage power during standby, and Figure 2.1 also illustrates various leakage components. Sub-threshold leakage [19], I_{sub}, is the dominant component, which occurs due to the weak inversion conduction when the gate-source voltage (V_{GS}) is less than the threshold voltage (V_T). Due to the drain induced barrier lowering (DIBL) effect, I_{sub} decreases exponentially with the reduction of the drain-to-source voltage (VDS). Gate

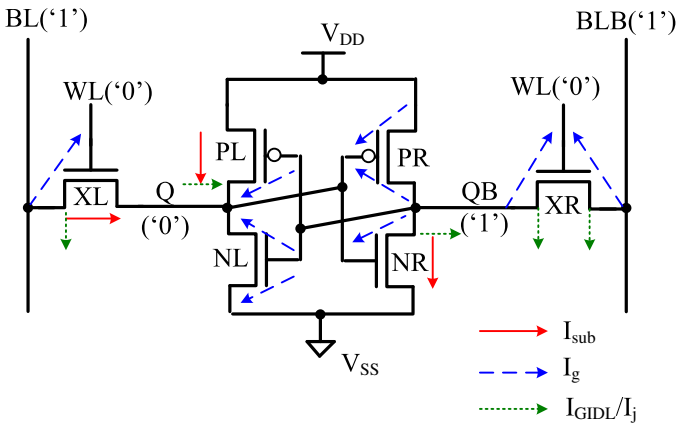


Figure 2.1: 6T SRAM cell and various leakage current paths inside the cell.

leakage [40], I_g, is the direct tunneling current through gate oxide to channel and the overlap region between gate and source/drain extention. It grows exponentially with the scaling of the gate oxide thickness and V_{DD} [40], although new high-k metal gate devices promise large reduction in gate leakage [42]. I_{GIDL} is gate induced drain leakage (GIDL) current, which is caused by the high electric field region under the gate-to-drain overlap region [66] and I_j is caused by the reverse-biased pn junction. Both I_{GIDL} and I_j decrease dramatically with V_{DD}. Therefore, V_{DD} scaling can effectively reduce the total leakage current of the cell, Ilk,total. Fig. 2.2 shows that $I_{lk,total}$ is reduced by more than 10$\times$ for a cell in 45nm. Due to the aspect of V_{DD} itself, the leakage power of the cell, which is equal to $I_{lk,total} \cdot V_{DD}$, is further reduced with a lower V_{DD}. Many designs have exploited this dependence on V_{DD} for SRAM leakage power reduction, by scaling down V_{DD} during standby and/or active operation [5, 18, 20, 29, 35, 51, 68].

However, collapsing V_{DD} degrades cell stability. Figure 2.3 shows how excessive V_{DD} scaling causes a bitcell to lose its original data ('0' in this example). Figure 2.3(a) shows

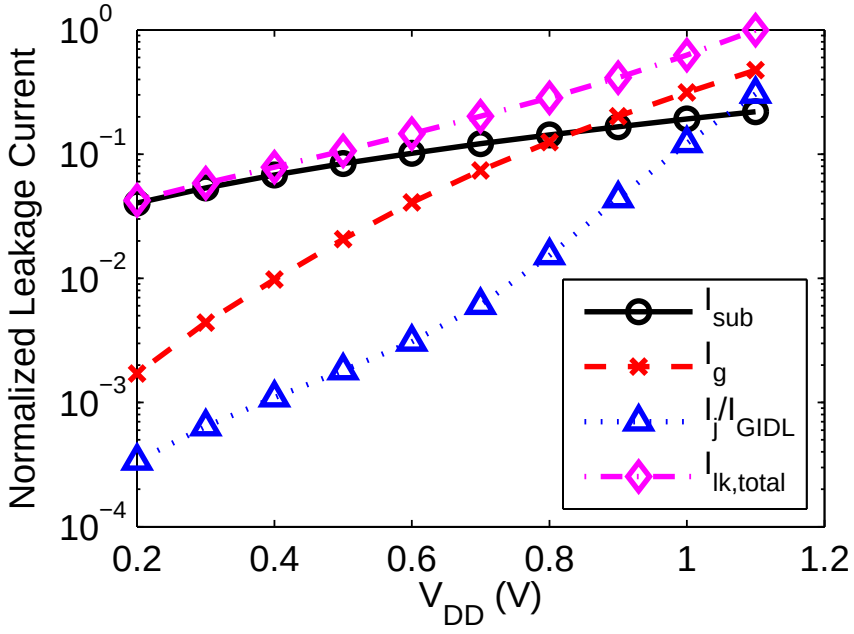


Figure 2.2: The normalized cell leakage current vs. V_{DD}.

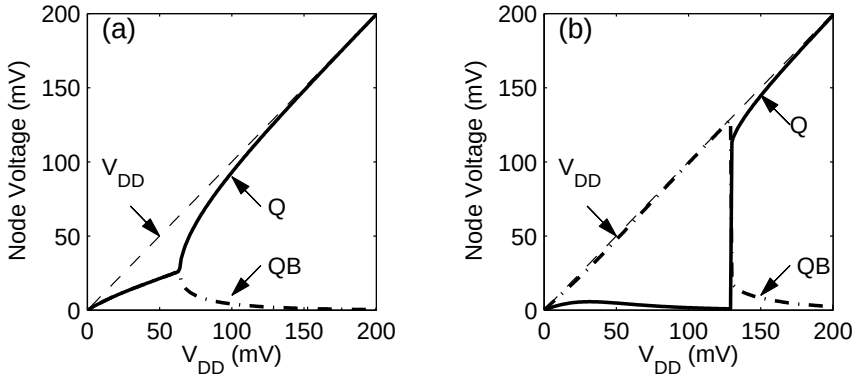


Figure 2.3: Cell nodes Q and QB (a) converge if the cell is balanced or (b) flip if the cell is imbalanced when V_{DD} is lowered than DRV.

the case when the cell is balanced (symmetric), with identical left and right halves. With V_{DD} scaling, the cell nodes Q and QB converge to a metastable point as a result of degraded gain, making the '0' and '1' states indistinguishable. Figure 2.3(b) shows the case when the cell is imbalanced by some variation induced mismatch in the transistors. In this case,

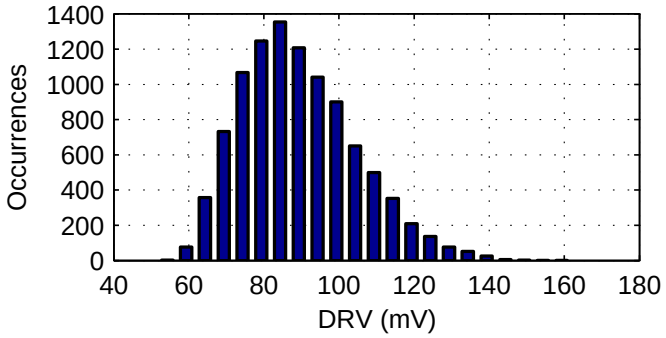


Figure 2.4: Simulated DRV histogram for a 10K-b SRAM in a 90nm node.

Q and QB flip to the more stable state ('1' here). The data retention voltage (DRV) defines the minimum V_{DD} that can be applied to an SRAM cell without losing data. Since a cell can store either a '0' or a '1', the actual DRV is computed as follows.

$$\text{DRV} = \max(\text{DRV}_0, \text{DRV}_1) \tag{2.1}$$

where DRV_0 is the DRV when the cell is storing a '0', and DRV_1 is the DRV when it is storing a '1'. If the cell is balanced, then $\text{DRV}_0 = \text{DRV}_1$. However, if there is any mismatch due to process variations, they become unequal. Particularly, one becomes much larger and the other becomes much smaller or even close to 0. Smaller DRV means the cell is more stable. This implies that mismatch will make the cell more stable for one data value while less stable at the other. Therefore, to obtain the real DRV, we must pick the worst (largest) from both DRV_0 and DRV_1.

We run Monte Carlo simulation with independent normally-distributed V_{T} variation on each transistor of the 6T cell. Figure 2.4 shows the histogram of a 10K-point MC simulation for the DRV of SRAM bitcells in a commercial 90nm CMOS process. The DRV exhibits a non-Gaussian distribution with a heavy tail on the right side. DRV values in this heavy tail

are most relevant for a fault-free SRAM array, since the worst cell in the array determines the minimum standby V_{DD} that can be applied to the array. Accurate estimation of the DRV values in the tail is essential for optimizing the trade-off between SRAM yield and standby power savings. If the tail value is over-estimated, the cell will be over-designed or over-protected thus limit the standby power savings. On the contrary, if it is under-estimated, more cell failures than predicted will occur and the yield cannot be met. However, for large memories, we need to estimate extremely rare DRV quantiles, and standard Monte Carlo simulation is too expensive, computationally. This problem motivates our work.

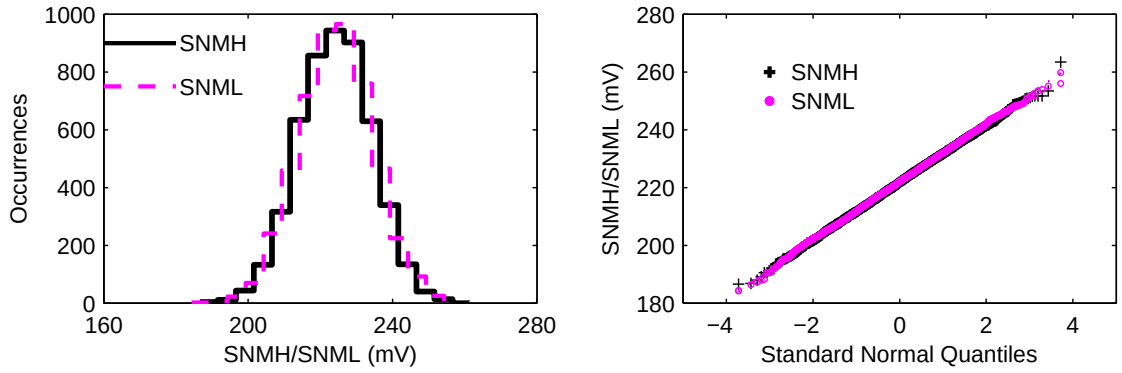
Since DRV is the minimum V_{DD} below which a cell can not preserve its data, we can also consider it as the V_{DD} at which static noise margin (SNM) first equals zero in a noise-less system. Therefore, we propose to use SNM as a starting point to explore DRV statistics.

2.3 Cell Hold SNM Statistics and Its Sensitivity to V_{DD}

SNM measures the amount of DC voltage noise that a cell can tolerate. SNM equals the length of the largest square that can be embedded between the voltage characterization curves (VTCs) of the two half-cells [55] as shown in Figure 2.6. Particularly, the largest square in the upper-left lobe is the SNM1, the SNM when the cell is storing a '0'; the largest square in the lower-right lobe is the SNM0, the SNM when the cell is storing a '1'. The actual SNM is computed as follows.

$$\text{SNM} = \min(\text{SNM1}, \text{SNM0}) \tag{2.2}$$

Within-die V_T variation is the major source of cell imbalance, thus it also has a huge impact on SNM. Figure 2.5(a) plots 5K-point samples of SNM1 and SNM0 from Monte



(a) (b)

Figure 2.5: (a) Histogram of SNM1 and SNM0 from 5K samples of MC simulation and (b) Quantiles of SNM1 and SNM0 vs the theoretical standard normal quantiles when V_{DD}=0.6V. SNM1 and SNM0 can be approximated with the same normal distribution.

Carlo simulations with normally distributed within-die V_T variation when V_{DD}=0.6V. The data show that the distributions of SNM1 and SNM0 are nearly identical and close to a normal distribution. If we further plot the SNM1/SNM0 quantiles versus theoretical standard normal quantiles in the quantile-quantile (Q-Q) plot (seen in Figure 2.5(b)), the points are roughly on a straight line, which implies that both SNM1 and SNM0 can be well approximated by a normal distribution. Therefore, we can accurately estimate the mean (μ) and standard deviation (σ) of SNM1/SNM0 by fitting a normal distribution to data from a small-scale (e.g. 1.5K$\sim$5K points) MC simulation.

Figure 2.6 shows the change of VTCs and the embedded SNM squares as we decrease V_{DD} using the same example bitcells as in Figure 2.3. Figure 2.6(a) shows that symmetry allows the cell to remain bistable to lower V_{DD}. It becomes clear that the DRV equals the supply voltage at which SNM is equal to zero in a noiseless system. Both SNM1 and SNM0 decrease symmetrically to zero, implying that DRV_0 and DRV_1 are equal. However, the stability of the imbalanced cell (and its DRV) strongly depends on the data pattern. For

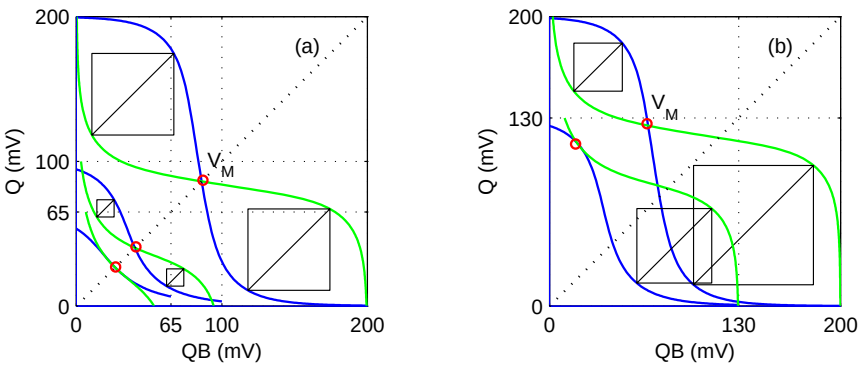


Figure 2.6: VTCs of (a) balanced and (b) imbalanced cells with varying V_{DD}; V_M is the trip point of the VTCs.

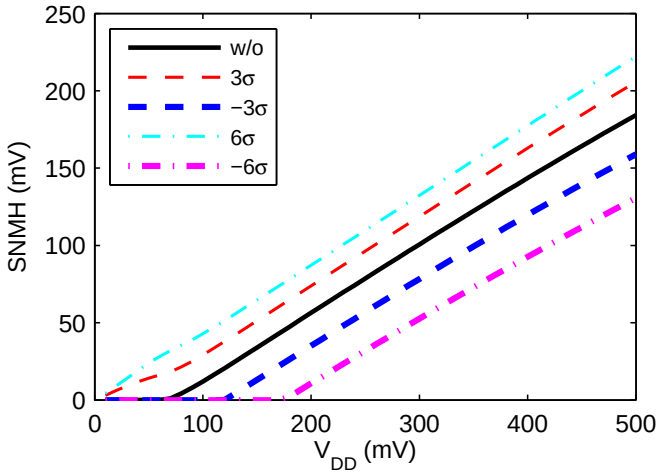


Figure 2.7: SNM vs V_{DD} under various mismatch scenarios.

example, in Figure 2.6(b) this particular imbalanced cell always has a larger SNM0, and its SNM1 decreases to zero at a lower V_{DD}. Therefore, this imbalanced cell is more sensitive to V_{DD} when Q=0, and its DRV is set by DRV_0.

To find DRV, we must lower V_{DD} until SNM reaches zero. So it is necessary to examine the sensitivity of SNM to V_{DD}. Figure 2.7 shows the simulated SNM1 vs V_{DD} under different mismatch conditions, including no V_T mismatch and $\pm 3/\pm 6\sigma$ V_T mismatch on

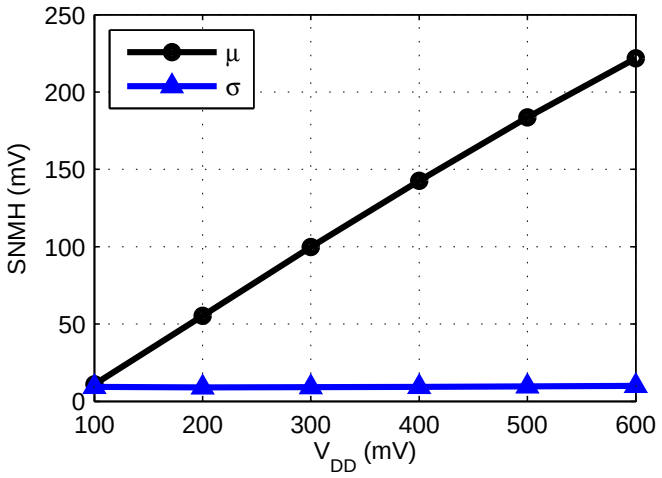


Figure 2.8: SNM vs V_{DD} under various mismatch scenarios.

NL (in Figure 2.1). In each case, the SNM varies linearly with V_{DD} with nearly identical slope. Based on these observations, we assume the sensitivity of SNM1 to V_{DD} is a constant k regardless of V_T mismatch on the cell transistors. SNM1 at voltage v, SNM1_v, can be approximated with (2.3):

$$\text{SNM1}_v = k(v - v_0) + \text{SNM1}_{v_0} \tag{2.3}$$

where v_0 is the initial V_{DD} and SNM1_{v_0} is the SNM1 value when V_{DD}=v_0.

Figure 2.8 shows the simulated SNM1 μ and σ from 5-K MC samples under different V_{DD}s. This plot verifies that σ remains constant while μ linearly deceases with V_{DD} scaling. This is reasonable since the shape of the distribution is mainly determined by the intrinsic parametric V_T variation, which is unchanged with V_{DD} scaling. The sensitivities of μ and σ to V_{DD} are:

$$\frac{\partial \sigma}{\partial V_{DD}} \approx 0, \quad \frac{\partial \mu}{\partial V_{DD}} \approx k.$$

Note that here we do not consider aging related mechanisms (e.g. NBTI) that change

device V_T after silicon implementation. To extract the linear fitting coefficient k, ideally we should run Monte Carlo simulations on multiple V_{DD} points and then fit a linear model to the curve of the mean of SNM1 versus V_{DD} as shown in Figure 2.8. Though this method is more accurate, it requires thousands of Monte Carlo simulations for each V_{DD}. We observe that the curve of the nominal SNM1 versus V_{DD} (the solid one in Figure 2.7) has about the same slope as the curve of the μ values versus V_{DD} in Figure 2.8. Based on this, we propose to extract k from the linear fit to the curve of the nominal results, which only requires a single short DC-sweep SPICE simulation and thus offers a great speedup. However, this simplified procedure might lead to some errors. We will further discuss how to maintain a good accuracy when using k extracted from a single DC simulation in Section 2.6.

2.4 SNM and DRV Statistical Model

The real SNM of the cell is the minimum of SNM1 and SNM0. As shown in Figure 2.5, the distribution of SNM1 and SHNL are almost identical because of the symmetry of the 6T cell. If we assume SNM1 and SNM0 are also independent random variables, then the cumulative density function (CDF) of SNM at supply voltage v is

$$\begin{aligned} F_{\text{SNM}_v}(s) &= \text{P}(\text{SNM}_v < s) \\ &= \text{P}(\min(\text{SNM1}_v, \text{SNM0}_v) < s) \\ &= \text{P}(\text{SNM1}_v < s) + \text{P}(\text{SNM1}_v \geq s, \text{SNM0}_v < s) \\ &= 2F_{\text{SNM1}_v} - F^2_{\text{SNM1}_v}, \end{aligned} \tag{2.4}$$

where $F_{\text{SNM1}_v} \approx \mathcal{N}(\mu, \sigma)$.

A cell hold failure occurs when the cell's SNM is less than s, which is often a positive

value. If more noise immunity is needed (e.g. for soft-error protection), s should be larger. We define $P_f(v, s)$ as the cell hold failure probability when $\mathrm{V_{DD}} = v$ and the minimum acceptable noise margin is s, which we compute in (2.5), using (3.3).

$$\begin{aligned} P_f(v, s) &= \mathrm{P}(\mathrm{SNM}_v < s) \\ &= \mathrm{erfc}(x) - \frac{1}{4}\mathrm{erfc}^2(x), \\ &\text{where} \quad x = \frac{\mu_0 + k(v - v_0) - s}{\sqrt{2}\sigma_0}, \end{aligned} \tag{2.5}$$

erfc(·) is the complementary error function, which can be computed numerically, and μ_0 and σ_0 are some estimates of the mean and standard deviation of $\mathrm{SNM1}_{v_0}$.

DRV is the minimum operation voltage during standby mode. More specifically, we denote the random variable DRV_s as the cell DRV for a specific noise margin requirement s. Thus, the failure of the cell at the supply voltage v can also be defined as the event when DRV_s is larger than v.

$$P_f(v, s) = \mathrm{P}(\mathrm{DRV}_s > v) \tag{2.6}$$

By equalizing (2.5) and (2.6), we can compute the inverse CDF of DRV_s as follows.

$$F^{-1}_{\mathrm{DRV}_s}(p) = \frac{1}{k}\left(\sqrt{2}\sigma_0 \cdot \mathrm{erfc}^{-1}\left(2 - 2\sqrt{p}\right) - \mu_0 + s\right) + v_0, \tag{2.7}$$

$$\text{where } \mathrm{P}(\mathrm{DRV}_s \leq F^{-1}_{\mathrm{DRV}_s}(p)) = p$$

and $\mathrm{erfc}^{-1}(\cdot)$ is the inverse function of erfc(·). (2.7) allows us to directly compute the standby supply voltage required to maintain a desired probability of hold failures.

A good estimation of the cell failure probability can improve the estimation of SRAM yield, which is a critical metric for SRAM design cost. Suppose we are designing an SRAM array with a capacity N (in bits) and the ability of tolerating up to R errors(e.g. by using

error correction, etc.). We can use (2.5) to estimate the cell failure probability p_f at one supply voltage. Then the SRAM yield, p_y, at this supply voltage can be computed from (2.8).

$$p_y = \sum_{j=0}^{R} \begin{pmatrix} N \\ j \end{pmatrix} p_f^j (1-p_f)^{N-j} \tag{2.8}$$

$$p_f = 1 - \sqrt[N]{p_y}, \quad \text{when} \quad R = 0 \tag{2.9}$$

On the other hand, for a given yield constraint, we can obtain the required cell failure probability by transforming (2.8). For simplicity, here we only consider the case when no errors are allowed (i.e. R=0). Then the required p_f can be computed from (2.9), and we can quickly predict the minimum V_{DD} value (i.e. DRV_s) satisfying this yield constraint from (2.7) with $p = 1 - p_f$.

Both (2.5) and (2.7) only require four parameters. They are the initial V_{DD} value (v_0), and other three fitting coefficients: the mean and standard deviation of SNM1 at v_0 (μ_0 and σ_0) as well as the sensitivity of SNM1 to V_{DD} (k). Now let us summarize the steps of using our model as followings:

1) Pick a value for v_0.

2) Extract μ_0 and σ_0 from a 1.5K$\sim$5K-point MC simulation of SNM1 when $V_{DD} = v_0$.

3) Extract k from a short DC-sweep of the nominal SNM1 vs V_{DD}.

4) Pick a value for s as a minimum acceptable noise margin.

5) Use (2.5) to compute the cell hold failure probability $P_f(v, s)$ when $V_{DD} = v$ and use (2.8) to compute the SRAM yield. *OR*

6) Use (2.7) to compute the minimum V_{DD} that can ensure the cell hold failure probability p_f (or the required yield from (2.9)).

We will discuss the sensitivity of the model to the four parameters in Section 2.6.

2.5 Model Evaluation

Monte-Carlo simulation of phenomena such as array-wide DRV can take huge amounts of time. As the memory size increases and samples from far out the tail are required, this simulation delay becomes untenable. Our new Statistical Blockade (SB) tool [3] improves upon traditional M-C for simulating rare events. To reduce simulation time, the Blockade tool classifies the possible M-C samples prior to simulation and selects only a subset of them that are likely to appear on the tail for simulation. After simulating this subset of points, the tool identifies the true tail points and uses them to fit a Generalized Pareto Distribution (GPD) model to the tail [3]. This statistical model allows estimation of events even farther out in the tail of the distribution of interest. In the next section, we show how we used the SB tool to verify the statistical DRV model and how the GPD model produced by the tool closely matches the actual DRV distribution.

We now test our DRV statistical model with comparison of the result from different methods. This experiment uses an SRAM cell implemented in an industrial 90 nm process.

Without loss of generality, we choose zero noise margin (i.e. $s = 0$) as the cell failure criterion. Figure 2.9 plots the DRV quantiles against the quantiles of a standard normal distribution. That is, if the q-th quantile of the standard normal distribution is equal to m (i.e. m-σ point for a standard normal) and the q-th quantile of the DRV distribution is equal to y, we plot the point at (m, y) coordinates of the figure. Since SRAM arrays

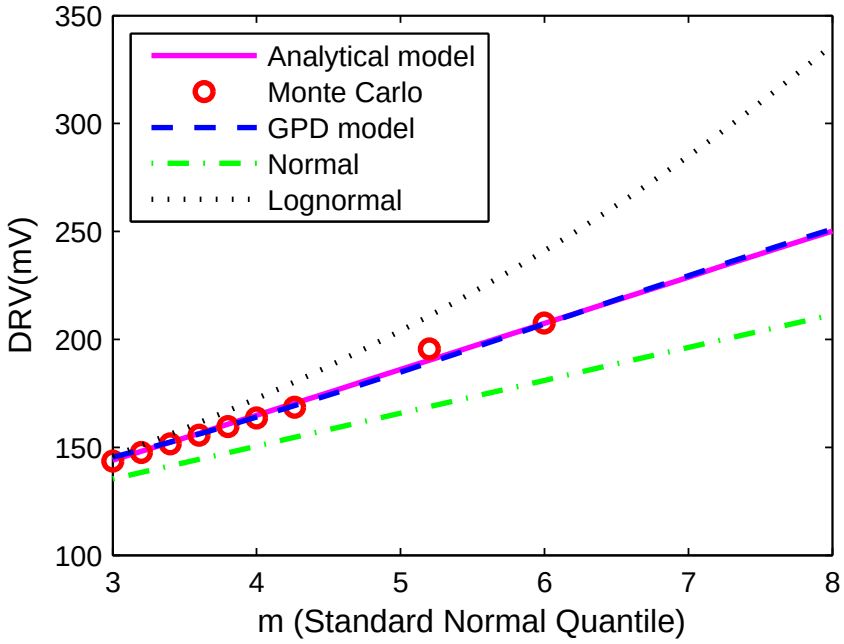


Figure 2.9: Estimates of DRV quantiles from five estimation methods. The GPD model closely fits the analytical model (2.7). The (red) circles with mle4 are obtained from standard Monte Carlo simulation and the 3 circles with m>4 show the worst DRV values from the three recursion stages of statistical blockade sampling. The normal and lognormal models are quite inaccurate.

usually have at least 1,000 bitcells, we are only interested in the quantiles larger than 99.9-th percentile, which is $\sim 3\sigma$ point of the standard normal distribution. Figure 2.9 uses five different methods to estimate the DRV quantiles for $m \in [3, 8]$:

1) *Analytical DRV model*: Use (2.7) with p equal to the probability of the normal quantile at m (i.e. $p = 0.5\text{erfc}(-m/\sqrt{2})$). We extracted k=0.425 from a DC sweep simulation for SNM1. We selected 100mV as v_0 and obtained the parameters μ_0=11.0mV and σ_0=9.3mV from a 5K-point Monte Carlo simulation.

2) *Standard Monte Carlo or fast Monte Carlo with recursive statistical blockade*: Use standard Monte Carlo for estimations below 4σ. Estimates greater than 4σ were

obtained by the recursive blockade method, thus allowing dramatically reduced simulation time. Note that here we are not using the GPD model, but only the empirical estimate from the sampled values. We obtain the results for total sample sizes of 100,000, 10 million and 1 billion Monte Carlo points from the recursive SB tool. The corresponding worst DRV value are estimates of the 4.26σ, 5.2σ and 6σ points, respectively.

3) *GPD model from recursive statistical blockade*: The 1,000 tail points from the last recursion stage of the recursive statistical blockade run are used to fit a GPD model, which is then used to predict the DRV quantiles.

4) *Normal*: A normal distribution is fit to data from a 1,000 point Monte Carlo run, and used to predict the DRV quantiles.

5) *Lognormal*: A lognormal distribution is fit to the same set of 1,000 Monte Carlo points, and used for the predictions.

From the plots in Figure 2.9, we can immediately see that the results from both the analytical DRV model and the GPD model closely track the Monte Carlo results up to 6σ. In addition, the two models match each other even at the $7\sim8\sigma$ tail. This matching of independently derived models increases the confidence that they are correct. It is also obvious that the normal and lognormal approximations are quite inaccurate for the DRV estimation in the tails. The normal fit is unable to capture the skewness of the DRV distribution, thus, underestimating DRV tail points. On the other hand, the lognormal distribution has a heavier tail than the DRV distribution and, thus, overestimates DRV tail points. Note here that, since our final GPD model uses t as the 99.9999-th percentile point ($\sim 4.75\sigma$ point), the

model does not have a real probabilistic meaning below $m = 4.75$. However, it can still be employed as a purely shape fitting function, as long as we are far from the mode in the distribution of DRV.

Our statistical DRV model provides significant speedup compared with Monte Carlo for the rare tail points of the DRV distribution. No matter how rare the event is, the DRV model only needs a few thousand MC simulations to extract μ and σ of the SNM1 distribution. It can thus provide a speedup up to $10^5\times$ over MC for a 6σ point, which requires at least 1 billion simulations.

2.6 Confidence in the Estimates

Intuitively, the statistical confidence of our estimates decreases as we predict farther out in the tail. In other words, the variance of the predictions will probably increase as we move out in the tail. Next, we will assess the confidence interval of the DRV estimation from the analytical DRV model, the GPD model as well as the Monte Carlo mothed.

Suppose we have n estimates $y_i(m), i = 1, \ldots, n$ for the $m\sigma$ point, say by building the statistical DRV model from n different Monte Carlo runs of SNM. From these estimates we can empirically compute the 97.5% percentile and 2.5% percentile points, $y_{97.5\%}(m)$ and $y_{2.5\%}(m)$, respectively. A 95% confidence interval $\kappa_{95\%}(m)$ can then be estimated as

$$\kappa_{95\%}(m) = y_{97.5\%}(m) - y_{2.5\%}(m). \tag{2.10}$$

The 95% confidence interval can also be expressed as $[\hat{y} - \alpha\hat{y}, \hat{y} + \alpha\hat{y}]$, where α is the radius of 95% confidence interval as a percentage of the mean of the estimates. Smaller α

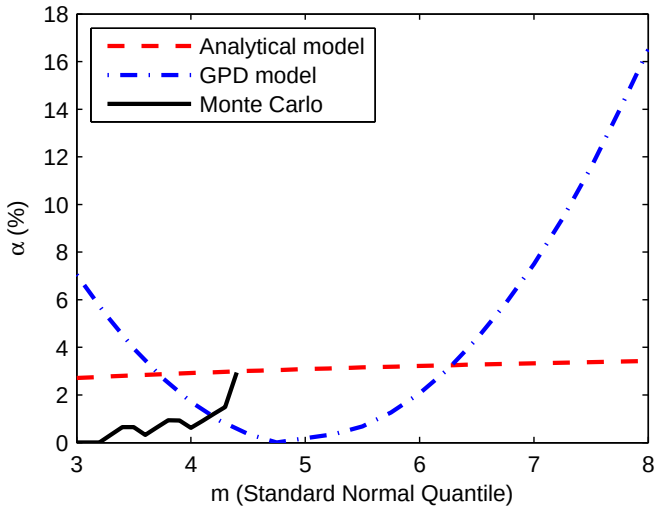


Figure 2.10: Radius of 95% confidence intervals as a percentage of the mean value for DRV estimation for the analytical model, the GPD model and the Monte Carlo method

implies lower variance in the DRV estimate. For a $m\sigma$ point, we then compute α as

$$\alpha(m) = \frac{\kappa_{95\%}(m)}{\frac{2}{n}\sum_{i=1}^{n} y_i(m)}. \tag{2.11}$$

We use this empirical method to compute 95% confidence intervals for the estimate of the $m\sigma$ point of the SRAM cell DRV, where $m \in [3, 8]$. Figure 2.10 shows the radius of the 95% confidence interval for the analytical DRV model, the GPD model and the Monte Carlo method (using 1 million sample points). Next, we will describe how we obtain the results for each approach.

2.6.1 Confidence Interval of the Analytical Model

The analytical model in (2.7) only requires four parameters: $k, v_0, \mu_0,$ and σ_0. The variance of the estimation is determined by the sensitivity of the model to these parameters. Since μ_0 and σ_0 are fitting coefficients from a small-scale Monte Carlo simulation of SNM1

when $V_{DD} = v_0$, we first assess the variance of the DRV estimates from the variance of these two parameters.

We can empirically estimate the confidence or variance of this model as follows: We first fix v_0 and k, and run n runs of MC simulations with n_{MC} samples each. That will give us n different pairs of μ_0 and σ_0. Then we use (2.7) to compute n estimates of DRV at the m point, and use (2.10) and (2.11) to compute the tightness of 95% confidence interval, $\alpha(m)$, under this pair of (v_0,k). We choose v_0=100mV and run 50 MC iterations using 1,000 samples for each. k is approximated to 0.425 by fitting the linear curve to the data from the DC simulation of the nominal SNM1 vs V_{DD} as the solid curve in Figure2.7. The dashed curve in Figure 2.10 shows the computed $\alpha(m)$ values, which are all below 4% for $m \in [3, 8]$.

We then use this method to compute $\alpha(m)$ for different pairs of (v_0,k) to check the sensitivity of the variance to v_0 and k. We alter v_0 from 100mV to 200mV and 300mV. For k, besides the value 0.425 from the nominal SNM1 vs V_{DD}, we evaluate another value, 0.4438, which is obtained from the linear fit to the estimated mean of SNM1 from those Monte Carlo samples at the three v_0 points. As we mentioned in Section III-A, it is much faster to obtain k from the nominal SNM1 by just running a single short DC sweep simulation. However, it is necessary to examine that whether this faster method can also offer the comparable accuracy. Here, we first compare the two k values in terms of the variance of the estimate. In section V-B-3, we will also show the accuracy of the mean of the estimate with these two k values. Figure 2.11 plots the radius of the 95% confidence interval for each pair of (v_0,k). For all the curves, although the statistical error (α) slightly increases with m, it remains within 4% (i.e. above 96% accuracy for 95% confidence interval) up to

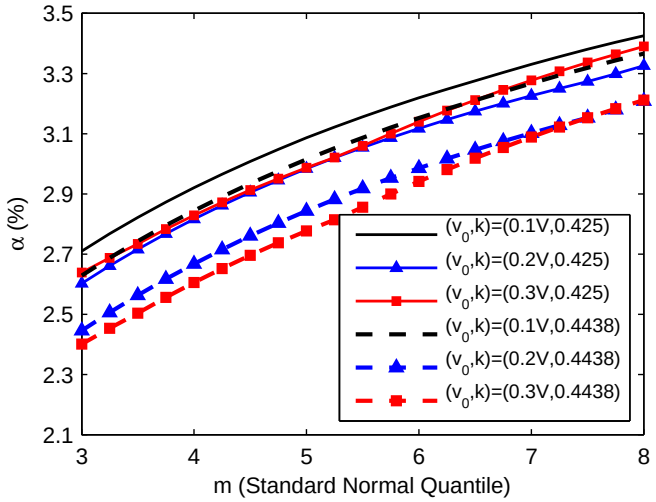


Figure 2.11: Radius of 95% confidence intervals as a percentage of the mean value for DRV estimation using the analytical model with different pairs of v_0 and k.

the 8σ point. This indicates that the variance of the DRV estimates from our model is not sensitive to either parameter.

2.6.2 Confidence Interval of the GPD Model & Standard Monte Carlo

Suppose that $x_i, i = 1, 2, \ldots, n$ are the sampled order statistics (e.g., DRV values); i.e., the sampled values *sorted* in ascending order. Then, any x_i is also an estimate of the i/n-th quantile. But, there is a non-trivial probability that some other order statistic $x_j, j \neq i$, may match the actual i/n-th quantile. Now suppose that p_t is such that np_t is an integer. Then, the probability that the i-th order statistic, x_i, equals the p_t quantile is given by a binomial distribution, which for large n can be well approximated by the normal distribution:

$$P(x_i = p_t\text{-th quantile}) \sim N(np_t, np_t(1 - p_t)). \tag{2.12}$$

A $\pm 2\sigma$ 95.45% confidence interval in terms of the quantile estimate index i is then given by

$$[l, h] = \left[\lfloor np_t - 2\sqrt{np_t(1-p_t)} \rfloor, \lceil np_t + 2\sqrt{np_t(1-p_t)} \rceil \right]. \tag{2.13}$$

We sample 10,000 pairs of GPD parameters from the joint normal distribution with this mean vector and its covariance matrix to compute different estimates of DRV. With these DRV estimates, we compute the confidence interval-based accuracy measure, α, using (2.10) and (2.11). The dash-dotted curve in Figure 2.10 shows the result of applying this method. It suggests that for error within 5% with a confidence of 95% we can predict out to 6.6σ (1 in 48.6 billion).

Finally, we compare the confidence intervals of the estimates from our two methods with the confidence intervals of standard Monte Carlo estimates. The confidence interval of the $m\sigma$ from an n-point Monte Carlo run is given by $[x_l, x_h]$, where l and h are given by (2.13), but now we replace p_t with the CDF associated with m: $\Phi(m)$. Once again, x_l and x_h are the l-th and h-th order statistics from the n-point sample. We now estimate confidence intervals for our 1 million point Monte Carlo run. The result is plotted as the solid curve in Figure 2.10. The width of the confidence interval from Monte Carlo increases as we estimate further out to the larger m points. Note that with 1 million MC samples, we can only obtain the confidence interval up to $\sim 4.5\sigma$ because there is no available DRV estimate beyond that.

We further compare the mean of the DRV estimation from the two models with Monte Carlo. Figure 2.12 shows the error relative to the result of Monte Carlo when $m \in [3, 4.25]$. Although we use the term error, it should be noted that the Monte Carlo estimate itself has some statistical error (Figure 2.10). For m$\leq$4, the GPD model offers less than 1% error.

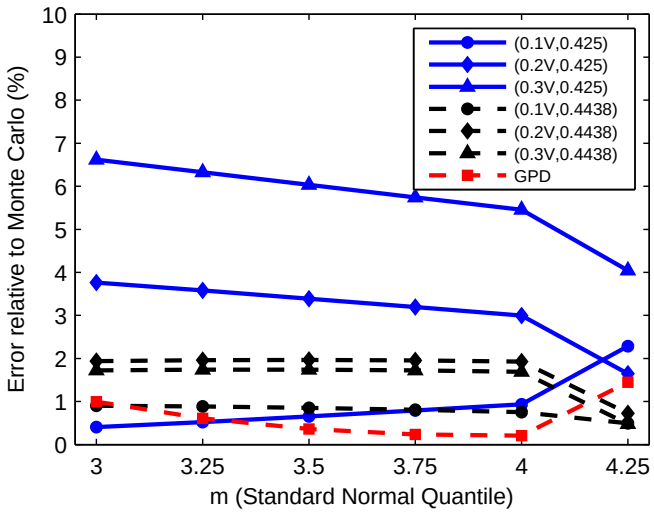


Figure 2.12: Error of the mean of the DRV estimates from the analytical model (the solid curves) and the GPD model (the dashed curve) relative to a 1 million point Monte Carlo. For the analytical model, results from different pairs of (v_0,k) indicate that a higher accuracy can be achieved by choosing v_0 near the DRV of an ideal cell, which is ∼100mV for this 90nm test case.

A slightly larger error occurs when m=4.25, where the Monte Carlo result is itself less confident as shown in Figure 2.10. In Figure 2.12, we also plot the error of the analytical model for different pairs of (v_0,k). Estimation with v_0=100mV shows the best agreement with Monte Carlo and the GPD model (∼1% of error for m≤4). It should be noted that the nominal DRV is less than 100mV for the 90nm node we used. With v_0=100mV, we can obtain more samples with negative SNM1/SNM0 values so that the approximated μ_0 and σ_0 can be closer to the true statistics of SNM1/SNHL, which are the key ingredients in the analytical DRV model. The results also show that when we use the k value 0.425 that is obtained from a single DC simulation of the nominal SNM1 vs V_{DD}, the accuracy of the DRV model is more sensitive to v_0. In this case, estimation with $v_0 > 100$mV shows relatively larger errors. However, if we choose k as 0.4438, the value from the linear fit to the curve of the mean of SNM1 vs V_{DD}, the sensitivity of the error to v_0 is reduced. This is

because the effect of different μ_0 at different v_0 is eliminated and the shift of the estimate is only caused by the relative small difference of σ_0 at distinct v_0 points. Therefore, we suggest that a value of v_0 close to, but larger than the DRV of an ideal cell is a better choice for a higher accuracy, especially when using k from the quick DC simulation of the nominal case for a faster estimate.

Chapter 3

Extended Statistical Method for SRAM Vmin and Yield Estimation

3.1 Motivation

In recent years, great efforts have been made to push down SRAM operational voltage to achieve more power savings in DVS environment ([3, 14, 28, 44]). Ultra-low V_{DD} SRAMs operating with V_{DD} near or below threshold voltage (VT) are also proposed for energy-efficient applications ([8, 61]). However, the minimum operational voltage (Vmin) is limited by the functionality of SRAM operations, including cell read stability, write ability, access speed, and hold stability. An accurate and fast prediction of Vmin and/or yield can accelerate the exploration of the potential tradeoffs among performance, power, and yield and contribute to design an optimum SRAM for a given application. However, the finding of the optimum Vmin becomes difficult in the presence of global and local variations.

In the previous chapter, we focus on Vmin during standby mode (i.e. the data retention voltage) for leakage power minimization. We derive a statistical model based on the sensitivity of the hold SNM distribution to V_{DD}. In this chapter, we further investigate active Vmin for read and write operations, which are essential for active power reduction and even more susceptible to variations under a low voltage. We discover that both cell read stability and write ability behave regularly with V_{DD} scaling as cell hold stability does. Thereby we propose a generic method to estimate Vmin for different operations. In addition, we generalize the form of the statistical model for both symmetrical and asymmetrical types of cells. To demonstrate the accuracy of our method, we compare it with both standard Monte Carlo and Importance Sampling.

Vmin is also determined by access failures due to an insufficient sensing signal developed within the required access time. In addition, the reliability issues such as negative bias temparature instability (NBTI) and soft gate oxide breakdown can cause Vmin drift. These topics are out of the scope of this paper. In this thesis, we focus on the impact of cell stability on Vmin in the presence of parametric variations.

The rest of the chapter is organized as follows. We first discuss read/write SNM metrics and their statistics with V_{DD} scaling. Then we present the details of our generic statistical model for Vmin and cell yield. Finally we describe the experimental setup and the experimental results with 6T SRAM in a commercial 45nm node.

3.2 Read and Write SNM Statistics

We denote SNM0 and SNM1 as the SNM when the cell stores ‘0’ and ‘1’. The true SNM is the minimum of SNM0 and SNM1. We also call the SNM for hold, read and write

operation HSNM, RSNM, and WSNM, respectively.

3.2.1 RSNM and WSNM Statistics at one V_{DD}

VTC-based read SNM has been widely used as the measure of SRAM read stability. The RSNM is the length of the largest square that can be embedded between the two VTC curves [55]. RSNM1 and RSNM0 measure the largest square inside the upper-left and lower-right lobe, respectively. Recently a new metric called N-curve has been proposed [69]. The stability of an SRAM cell is determined by I_{CRIT}, the peak current of the 'N curve'. [39] demonstrates a linear correlation between RSNM and I_{CRIT}. In this work, we choose the traditional RSNM as the metric to investigate read Vmin. The same methodology can be used for I_{CRIT}. Under normally distributed random parametric variation, both VTC-based RSNM and I_{CRIT}. Fig. 3.1 shows the quantile of RSNM0 and RSNM1 versus the theorical quantile of a standard normal variable. The near strict linearity between RSNM0/RSNM1 quantile and the standard normal quantile implies that each of them can be fitted to a normal distribution. In addition, the two distributions almost overlap with each other, which confirms that they are identical distributions.

Different methods have been used to measure write static noise margin. The first method is based on the VTC butterfly curves [4]. It measures the width of the smallest square that can be embedded between the VTC curves. The second criterion is measured by sweeping down the bitline voltage (VBL) [77]. The write margin is defined as the BL voltage at the point when the internal nodes flip. An alternative criterion measures the wordline voltage (VWL) [22]. In this case, the write margin is defined as the margin between V_{DD} and the WL voltage when the nodes flip. Instead of measuring voltages, a similar criterion

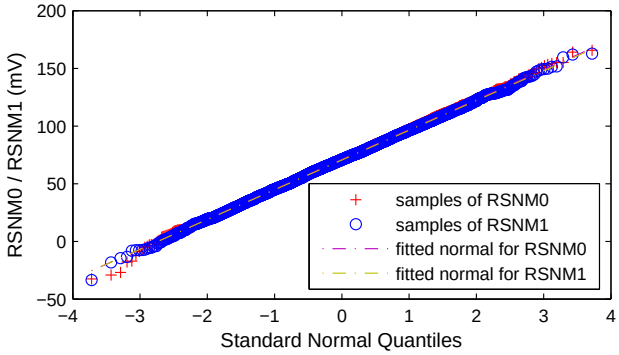


Figure 3.1: The quantile of RSNM0 / RSNM1 vs. the quantile of a theoretical standard normal variable. The near strict and equal linearity implies that they can be well fitted to the same normal distribution.

as I_{CRIT} based on the N-Curve approach is proposed for write ability [23]. We call it WTI. We have discovered that VWL, VBL and VTC metrics have stronger correlation with the real dynamic write margin [64]. Therefore, these three metrics are better candidates. We further examine the statistical characteristic of those metrics. Fig. 3.2 shows the Quantile-Quantile (Q-Q) plot of the normalized static write margin data using each method from simulation. Only the VBL and VWL data exhibit the nice linearity, which implies that they can be well approximated with a normal distribution. And interestingly, these two data almost overlap with each other. On the contrary, the VTC based and N-Curve based write margin distribution are skewed at either one or two ends. Since normal distribution can be well modeled, we prefer to use either the VBL or the VWL metrics as the write ability criterion. Here, we use the VWL metric to estimate Vmin. All the WSNM values we mention in this paper are from the VWL metric. But the same method can be applied if choosing VBL metric instead.

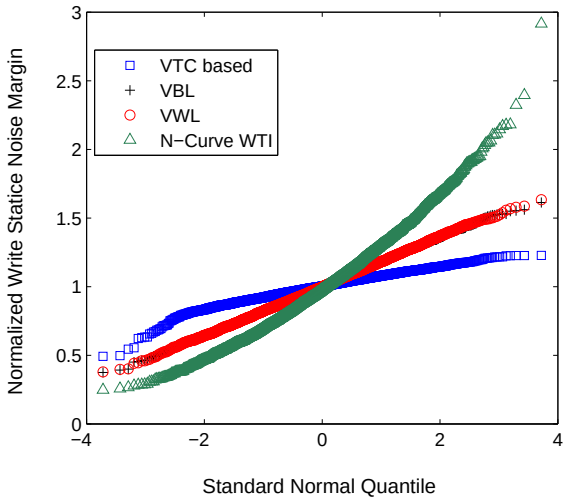


Figure 3.2: Q-Q plot of the normalized write static noise margin with different methods. The linearity of the bitline (VBL) and wordline (VWL) curves implies that they can be well approximated as a normal distribution.

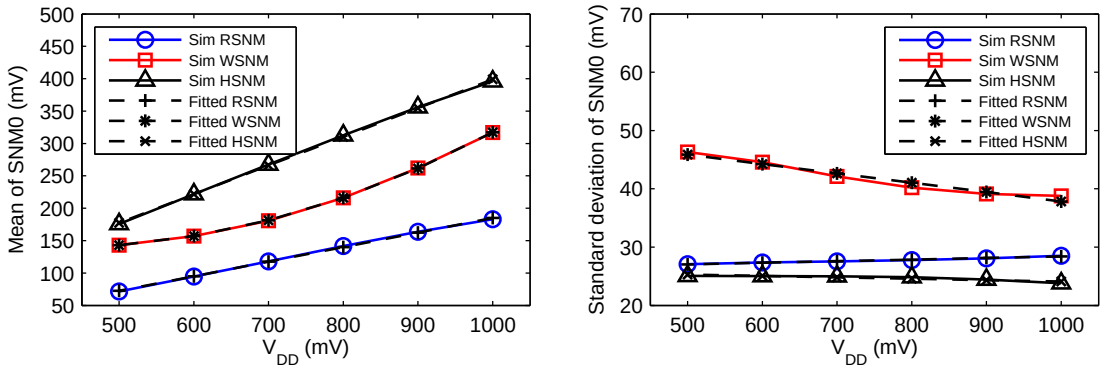
3.2.2 RSNM and WSNM Statistics at Arbitrary V_{DD}

We have demonstrated that RSNM0/RSNM1 and WSNM0/WSNM1 at one V_{DD} follow normal distribution. Since we are interested in the minimum operational voltage, an more important question is how those distributions would change with V_{DD} scaling.

We observe that all the distributions remain Gaussian after lowering V_{DD}. Moreover, as seen in Fig. 3.3, the sensitivity of the mean μ and the standard deviation σ of each SNM0 distribution to V_{DD} actually exhibits a nice trend, which can be fitted with the polynomial models as:

$$\frac{\partial \mu}{\partial V_{DD}} \approx a \cdot V_{DD} + b, \quad \frac{\partial \sigma}{\partial V_{DD}} \approx c. \tag{3.1}$$

Here, a, b, and c are fitting coefficients. For the technology and the SRAM cell we used, the fitting errors are no greater than 2.4% for all the curves. Now if we know the estimate of μ and σ of SNM0 at one initial supply voltage v_0 are μ_0 and σ_0, then we can compute μ



(a) (b)

Figure 3.3: (a) The mean value and (b) the standard deviation of the simulated RNM and WNM with the change of V_{DD}. The mean of SNM is fitted to the 2nd degree polynomial and the std of SNM is fitted to the 1st degree polynomial.

and σ of SNM0 at any new V_{DD}, v, as:

$$\mu = \mu_0 + a(v^2 - v_0^2) + b(v - v_0), \quad \sigma = \sigma_0 + c(v - v_0). \tag{3.2}$$

Note that for symmetrical cells such as the traditional 6T cell, the μ and σ of RSNM1/WSNM1 are equal to those of RSNM0/WSNM0; for asymmetrical cells like the novel 5T cell, their values might differ.

3.3 Vmin and Yield Model

In this section, we derive our generic Vmin model from the connection of Vmin and SNM with cell failure probability.

Let us first look at SNM. SNM_v is denoted as the cell SNM under a given supply voltage v. The failure of the cell occurs when its SNM_v is less than the minimum acceptable noise margin, say s. We denote $P_{cf}(v, s)$ as the cell failure probability for supply voltage v and

the acceptable NM s. It is equal to the value of the cumulative density function (CDF) of SNM_v at s:

$$\begin{aligned} \text{P}_{\text{cf}}(v,s) &= \text{P}(\text{SNM}_v < s) \\ &= \text{P}(\min(\text{SNM0}_v, \text{SNM1}_v) < s) \\ &= \text{P}(\text{SNM0}_v < s) + \text{P}(\text{SNM1}_v < s) \\ &\quad - \text{P}(\text{SNM0}_v < s, \text{SNM1}_v < s). \end{aligned} \tag{3.3}$$

For symmetrical cells like the conventional 6T cell, SNM0 and SNM1 are identical normal random variables. By applying this fact and assuming they are also independent variables, we can further obtain (3.4)

$$\begin{aligned} \text{P}_{\text{cf}}(v,s) &= 2F_{\text{SNM0}_v}(s) - F^2_{\text{SNM0}_v}(s) \\ &= \text{erfc}(x) - \frac{1}{4}\text{erfc}^2(x) \qquad (3.4) \\ \text{where} \quad x &= \frac{\mu - s}{\sqrt{2}\sigma} \qquad (3.5) \end{aligned}$$

where μ and σ are the mean and std of SNM0 at supply voltage v, and erfc($\cdot$) is the complementary error function, which can be computed numerically. Now by applying (3.2), x can be expressed as

$$x = \frac{\mu_0 + a(v^2 - v_0^2) + b(v - v_0) - s}{\sqrt{2}(\sigma_0 + c(v - v_0))} \tag{3.6}$$

This allows us to quickly estimate the cell failure probability at any V_{DD} without rerunning simulation at the new V_{DD}.

Besides SNM, an alternative way to compute cell failure probability is from Vmin. We denote Vmin_s as the cell's minimum operation voltage for an acceptable noise margin s. Thus, the failure of the cell at the supply voltage v can also be defined as the event when

Vmin_s is larger than v.

$$\text{P}_{\text{cf}}(v, s) = \text{P}(\text{Vmin}_s > v) \tag{3.7}$$

By equalizing (3.4) and (3.7), we can obtain x from (3.8) for a desired probability of cell failures p and then compute the required Vmin value, v, by solving (3.6).

$$x = \text{erfc}^{-1}(2 - 2\sqrt{1-p}) \tag{3.8}$$

here $\text{erfc}^{-1}(\cdot)$ is the inverse function of erfc(·).

So far we assume that the cell is symmetrical so that SNM1 has the same distribution as SNM0. In the case of an asymmetrical cell (e.g. 5T cell), SNM0 and SNM1 could be different normal variables. Suppose μ_l and σ_l are mean and std of SNM0, μ_h and σ_h are mean and std of SNM1.

$$\text{P}_{\text{cf}}(v, s) = 1 - \frac{1}{4}\text{erfc}(\frac{s-\mu_h}{\sqrt{2}\sigma_h}) \cdot \text{erfc}(\frac{s-\mu_l}{\sqrt{2}\sigma_l}) \tag{3.9}$$

$$\text{where} \quad \mu_h = \mu_{h,0} + a_h(v^2 - v_0^2) + b_h(v - v_0),$$

$$\sigma_h = \sigma_{h,0} + c_h(v - v_0),$$

$$\mu_l = \mu_{l,0} + a_l(v^2 - v_0^2) + b_l(v - v_0),$$

$$\sigma_l = \sigma_{l,0} + c_l(v - v_0)$$

For a given $\text{P}_{\text{cf}}(v, s)$, we can numerically solve (3.9) to obtain the required Vmin value, v.

3.4 Experimental Setup

We test our method with a 6T SRAM cell in a commercial 45nm CMOS technology. Without loss of generality, we choose 0 as the acceptable noise margin (i.e. s=0).

3.4.1 Vmin Simulation Methods

Since Vmin is the minimum V_{DD} for a non-negative SNM, we can find one cell's Vmin value through iterations of SNM simulations. For each iteration, we simulate SNM with the V_{DD} value decreased by one step until the SNM drops below 0. The drawback is, it costs many dc simulations for each Vmin sample. To reduce simulation time, we use an alternative method to simulate Vmin with a single dc run.

we connect the cell supply voltage, WL, and BL/BLB to V_{DD}. Then we do a dc simulation by sweeping down V_{DD}. The read Vmin ($Vmin_R$) is the V_{DD} value before Q and QB flip to the opposite state. Fig. 3.4(a) shows an example where $Vmin_R$=581mV. As illustrated in Fig. 3.4(b), using the SNM simulation method, we obtain the same $Vmin_R$ value, at which the RSNM0 (the square inside the lower lobe) becomes 0. Thus, a single dc sweep can replace multiple SNM simulations for finding $Vmin_R$.

For a write '1'('0') operation, we tie the cell supply voltage, WL and BL(BLB) to V_{DD} and BLB(BL) to ground; Q and QB initially are holding '0'('1') and '1'('0'). Then we perform one single dc sweep on V_{DD} from low to high. The write Vmin ($Vmin_W$) is the point where Q and QB start to flip. Fig. 3.5(a) shows an example where the cell has $Vmin_W$=512mV. Fig. 3.5(b) shows the results of the static write margin simulation for the same cell. WSNM is the margin between V_{DD} and the WL voltage where Q and QB flip. WSNM is 0 at V_{DD}=512mV, which equals to $Vmin_W$. Again, a single dc sweep can replace multiple write margin simulations to identify $Vmin_W$.

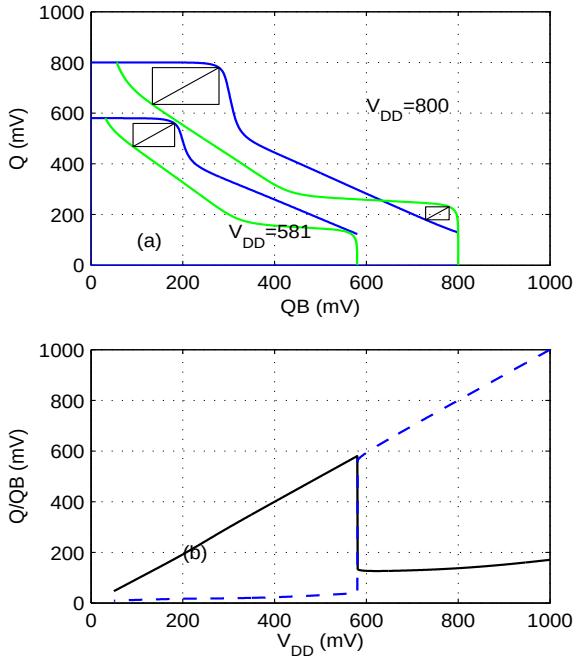


Figure 3.4: (a) For one V_{DD}, RSNM0 is the largest square that can be embedded between the two VTC curves; When V_{DD}=581mV, this cell has RSNM0=0; (b) Using alternative dc simulation setup, Q and QB cell nodes flip when V_{DD} <581mV. Thereby we can obtain the same V_{minR} value from the two simulation methods.

3.4.2 Importance Sampling

Importance Sampling (IS) is a widely used technique to reduce the variance of Monte Carlo simulation. Suppose parameter X has the original density $f(x)$ and the sampling density $g(x)$, and Y is the output of an unknown function of X. Then the estimator of the probability $p = P(Y > y)$ for some threshold y is given by

$$\hat{p}_(y) = \frac{1}{n}\sum_{i=1}^{n}\frac{f(X_i)}{g(X_i)}D(Y_i) \tag{3.10}$$

where n is the total number of samples and

$$D(Y_i) = \begin{cases} 1, & Y_i > y \\ 0, & Y_i \leq y \end{cases}.$$

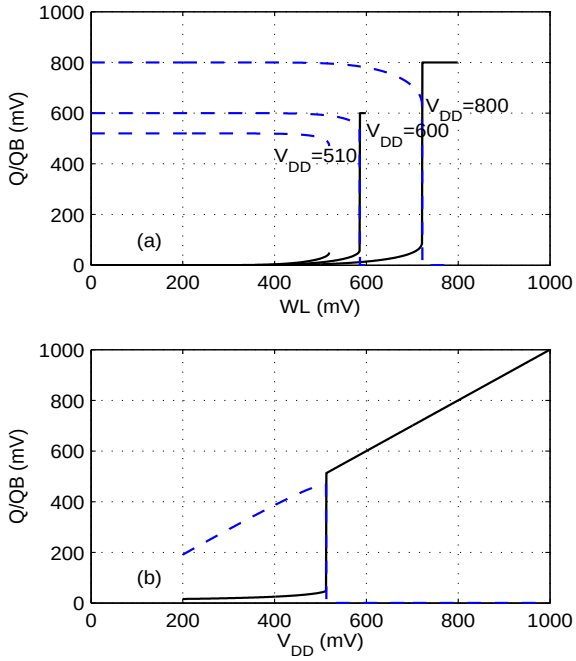


Figure 3.5: (a) For one V_{DD}, its write margin is obtained by sweeping WL voltage until Q and QB flip; When V_{DD}=512mV, this cell has 0 write margin; (b) Using alternative dc simulation setup, Q and QB cell nodes are unable to flip when V_{DD} <512mV. Thereby we can obtain the same V_{minW} value from the two simulation methods.

When $g(x) = f(x)$, (3.10) actually gives the standard Monte Carlo estimator.

For the application of SRAM, since V_T variation has the biggest impact on cell stability, we only modify the density of V_T for each transistor and assume they are independent random variables X_i, $i \in [1, 6]$. Originally, X_i is a random normal variable $X_i \sim N(\mu_i, \sigma_i^2)$. The key of IS is to choose a good sampling distribution that can efficiently generate rare events. [30] suggested using a mixture of shifted and ratioed distributions. Recently, [16] proposed to use a widened distribution. For simplicity, we choose the latter with $X_i \sim N(\mu_i, (\beta_i \sigma_i)^2)$, β_i=3.

An estimator of the empirical quantile ξ for $\theta = P(Y \leq \xi)$ is computed by [27] as:

$$\hat{\xi} = (\max\{y : \hat{p}(y) > 1 - \theta\} + \min\{y : \hat{p}(y) \leq 1 - \theta\})/2. \tag{3.11}$$

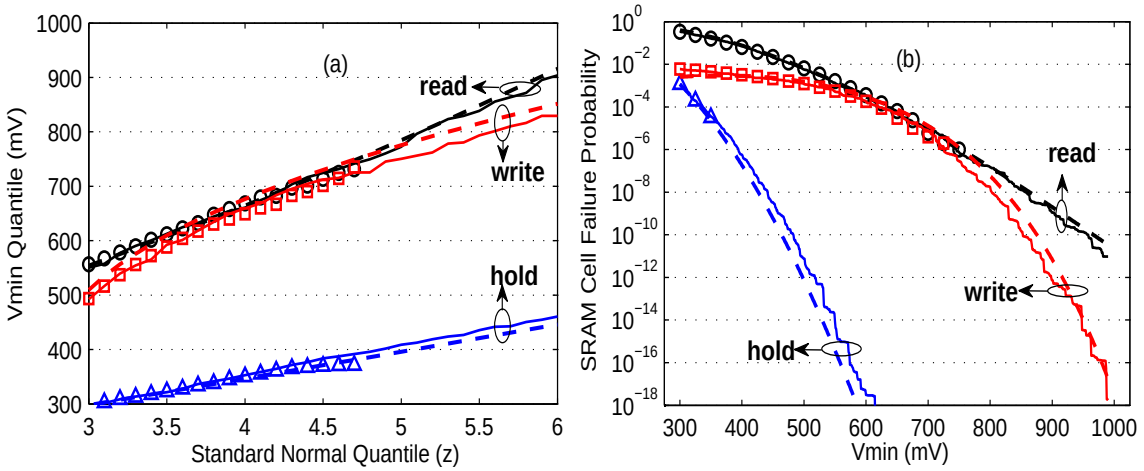


Figure 3.6: Estimates of (a) Vmin and (b) cell failure probability for read/write/hold from the theoretical models (3.8) and (2.5) (dashed curves) are compared with Monte Carlo (circles for read, squares for write, and triangles for hold) and Importance Sampling (solid curves).

3.5 Experiment Results

We now test our model with a commercial 6T SRAM cell in a 45nm technology. Without loss of generality, we choose zero noise margin (i.e. $s = 0$) as the cell failure threshold.

3.5.1 Vmin Estimation

Fig. 3.6(a) plots the Vmin quantiles against the quantiles of a theoretical standard normal variable X for read, write, and hold operation. For the point at the (x, y) coordinates of the figure, $P(\text{Vmin} \leq y) = P(X \leq x) = \Phi(x)$. Since SRAM arrays usually have at least 1,000 bitcells, we are only interested in the quantiles larger than the 99.9-th percentile, which is the $\sim 3\sigma$ point of a standard normal distribution. The results from the following three methods are plotted for each operation.

1) The Vmin estimates from (3.8) are plotted as the dashed curves.

2) The Vmin estimates from 1-million Monte Carlo simulations with the methods described in Section 3.4.1 are plotted with markers (circles for read, squares for write and triangles for hold). With 1-million samples, the maximum quantile we can obtain from MC is equivalent to the probability of the $\sim 4.7\sigma$ point of standard normal distribution.

3) The Vmin estimates from Importance Sampling are computed by (3.11) and plotted as solid curves. 50,000 MC samples with the new sampling distribution are simulated.

For tails within 4.7σ, the maximum error of the model and IS relative to the standard MC is 4.88% and 5.5% for each operation (seen Table.3.1). A relatively larger error occurs above 4.5σ. In fact, the MC result itself is less confident at this region because of few occurrences of the event. With more MC samples, the difference in this region might be even smaller. For the region beyond 5σ, MC is too costly, but the model shows a good agreement with IS. The consistency of the results from two independent methods increases the confidence of their accuracy.

Table 3.1: The maximum Vmin error relative to standard MC

	Read	Write	Hold
From model (3.8)	2.21%	4.88%	2.63%
From IS (3.11)	2.18%	2.51%	5.50%

With comparable accuracy, our model offers a huge speed-up relative to MC. For instance, if we want to design a 1M-b SRAM with 99% yield, the cell failure probability must be smaller than 1e-08 (5.612σ), i.e. 1 out of 100-million samples would fail. For a

full MC method, this requires at least 100-million runs. But our method only requires a small number of MC runs (e.g. 1$\sim$5k) for SNM at several typical V_{DD} points (e.g. 5$\sim$6) to obtain the coefficients in (3.1). Thus the speed-up of our method relative to MC can be at least $10^4\times$. Importance Sampling also gains a huge speed up over MC. However, using the simple scaled sampling distribution with 50k samples, its variance for the tails beyond 6σ is still large. To reduce variance, we have to either run more simulations with current sampling distribution or choose a more proper one and rerun simulations for a particular tail point. On the contrary, our method can estimate Vmin at any tail point (even $> 8\sigma$) with the same $\sim$30k (i.e. 5k$\times$6) SNM samples. We can compute the radius of 95% confidence interval of our model with the method described in Chapter 2.6. It remains $<$5.5% of the mean of the estimates for all the operations across the tail region up to 8σ.

3.5.2 Yield Estimation

With (2.5), we can quickly estimate the cell failure probability at any new V_{DD}. Fig. 3.6(b) plots the estimated cell failure probability from three methods for different operations. The results from our model show a good agreement with both Monte Carlo and Importance Sampling. With comparable accuracy, our method offers a significant acceleration for predicting the trend of the cell yield/failure probability across the whole V_{DD} range, which provides useful information for SRAM designers.

First, the designer can quickly tell that the yield of this cell is more limited by read operation. This can guide the designer to improve read stability in the early design phase. Fig. 3.6(b) also informs the designer that the gap between read failure probability (P_{rf}) and write failure probability (P_{wf}) varies with the operation voltage. For $V_{DD}>$880mV,

P_{wf} is at least 2 orders of magnitude smaller than P_{rf}. Thus, it is safe to only enable a read assist technique but to disable the write assist technique for saving extra power/performance overheads from write assist. When $V_{DD} \in$[650,800]mV, both the read and write operation are likely to fail at a moderate rate (¡1e-4). Thereby, we should turn on both the read and write assist features. However, when $V_{DD} <$650mV, both P_{rf} and P_{wf} are higher than 1e-4, which requires more effort to assist these operations. So either more voltage bias should be applied in the assist methods or other redundancy and/or repair techniques such as ECC and row/column replacement should be activated.

Overall, a quick and accurate estimation of the cell failure probability across all the possible operational voltage region can help designers find the best solution to improve yield but with the minimum cost in a shorter time.

Chapter 4

Canary-based Adaptive System for SRAM Standby Vmin Minimization

4.1 Motivation

With technology scaling, the leakage power consumed by transistors grows dramatically and becomes the most important challenge for many applications in both active and standby mode. For battery-constrained devices, the reduction of standby leakage power is especially important for longer battery life. Since SRAM/Cache is the largest component in many digital systems or SOCs, its leakage power during standby mode usually dominates the overall standby leakage power. Therefore, it is important to reduce SRAM standby leakage power. One of the most effective leakage reduction techniques is the scaling of supply voltage (V_{DD}). All the leakage current components, including sub-threshold leakage current, gate leakage, and junction leakage current, decrease dramatically with a smaller V_{DD}. Leakage power decreases even more rapidly due to the reduction of both V_{DD}

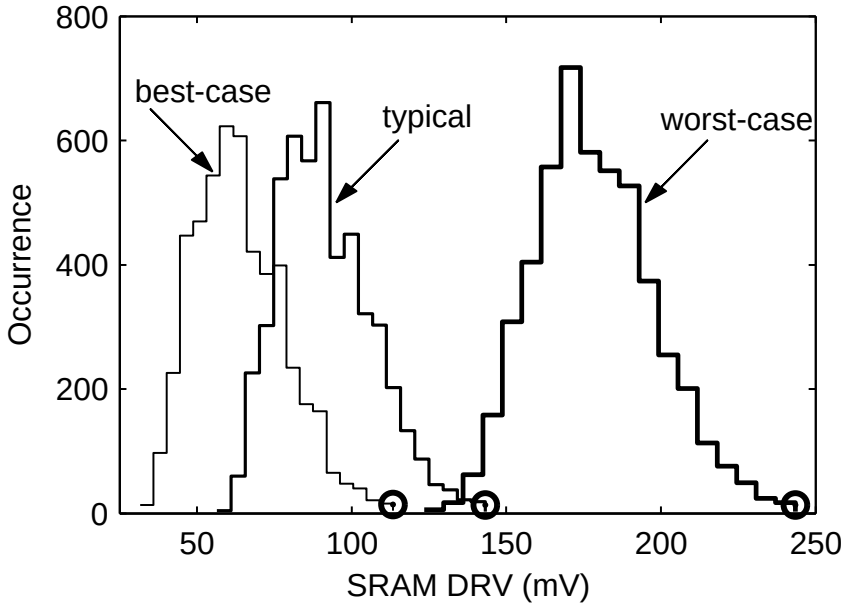


Figure 4.1: DRV distribution of a 5Kb SRAM array with global PVT variations and local variations. Three PVT cases (typical, best-case, and worst-case) are shown.

and leakage current. However, the minimum V_{DD} for the cell to preserve its data is limited by the data retention voltage (DRV). As discussed in Cahpter 2, local variation spreads the DRV of the cells across the chip. To preserve all the data in SRAM, V_{DD} must be above the standby Vmin, i.e. DRV of the worst cell within the SRAM array. In this chapter, we further discuss the impact of global variation on standby Vmin.

Global variations include the manufacture related process variations, voltage supply fluctuations, and temperature changes (i.e. PVT variations). We assume the temperature range is [0°C, 105°C] and the voltage fluctuation range is [-25mV, 25mV]. Fig. 4.1 shows the DRV histogram of a 5-Kb SRAM array at three PVT cases: typical, best-case, and worst-case. The typical case is at the TT (typical-N and typical-P) process corner, 25°C and 0 voltage fluctuation; the best case for the technology we use is at the SS (Slow-N and slow-P) process corner, 0°C, and 25mV voltage fluctuation; the worst case happens at the

FS (Fast-N and slow-P) process corner, 105°C, and -25mV voltage fluctuation. Under one PVT scenario, local variations spread the DRV of the cells, and the tail of the distribution (marked with circle) determines the standby Vmin for this global condition. In contrast, global variations predominantly move the entire DRV distribution around, so the tail point, i.e. the standby Vmin, also shifts with global effects. For this 90nm node, the worst-case Vmin ($Vmin_{wc}$) is about 100mV and 140mV higher than the typical case Vmin ($Vmin_{typ}$) and the best-case Vmin ($Vmin_{bc}$). For more advanced nodes, the variability of global effects might increase and result in a larger difference between $Vmin_{wc}$ and $Vmin_{typ}$/$Vmin_{bc}$. To ensure data safety under all the conditions, we must address this Vmin variability.

The most straightforward way to tolerate all the variations is the worst-case based open-loop approach. The designer selects a standby V_{DD} based on the worst case of the DRV under all the variations at design time and even adds an extra margin for more protection of the data in the cells. For instance, authors of the drowsy cache set the standby V_{DD} 50% higher than the threshold voltage despite the fact that the actual DRV can often be much smaller [5]. A processor with a drowsy mode is also implemented by collapsing the supply voltage well above the value that is required to upset the logic states during standby mode [6]. Although this worst-case open-loop approach is quite robust, it can relinquish substantial power savings because the full range of potential DRVs can be quite large when accounting for the worst case. The worst-case based approach wastes more power consumption under better scenarios as the variability increases with technology scaling.

Another way to address the Vmin variability is to reduce the range of the Vmin spread. The spread of DRV due to local variations can be reduced by optimizing the bitcell sizing. For example, using the longer length for the bitcell transistors can reduce the variation of

the threshold voltage due to the random dopant fluctuations [8]. Adaptive body biasing can reduce the Vmin differences cuased by process variations among dies. With the aid of redundancy and error correction techniques, Vmin can be further reduced to below the worst cell DRV [9]. However, these methods produce overhead cost and may degrade other important functional metrics. Even after attempts to reduce DRV variation, the open-loop worst-case approach can still over-protect the non-worst-cases and limit power savings, especially under large environmental changes.

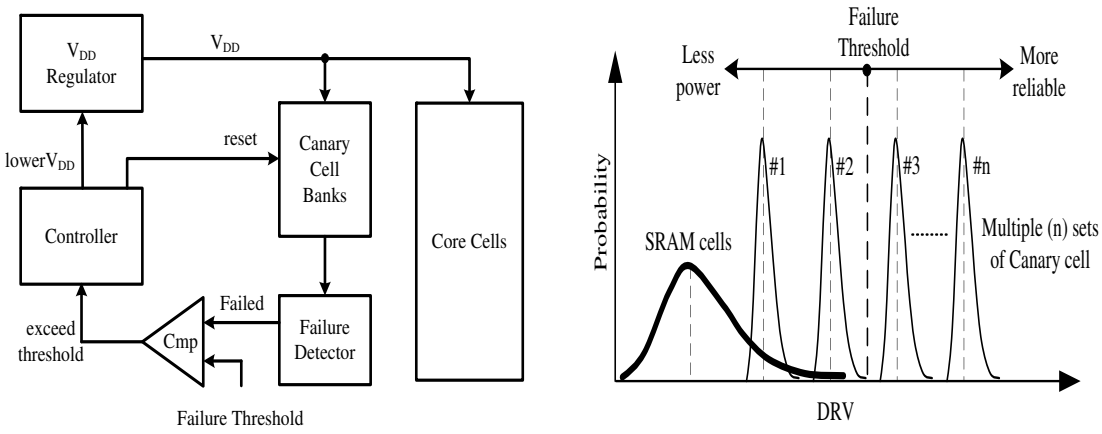
A more effective way to combat both physical and environmental variations is the adaptive approach, which adjusts circuits with the varying conditions. To save leakage power while maintaining data during standby, diode clamping based scheme is first proposed [48]. The cell bias is clamped at (V_{DD}-Vt). To enable more leakage power savings under process variation, [72] proposes to add a controllable resistor to dynamically control the clamped cell bias according to the measured leakage. Similarly, in [76], programmable bias transistors are used to tune the cell bias to compensate process skews. However, these passive methods have to add margins for the worst environmental and aging condition, which diminishes leakage power savings. voltage generator are proposed to improve PVT variation tolerance in standby operation [25, 32]. The reference voltage is programmed based on the actual standby Vmin under process variation and designed to insensitive to PVT conditions. Although this can effectively tolerate process and voltage fluctuation, it can not track the impact of temperature change on standby Vmin. SRAM data retention voltage actually increases with temperature. A fixed standby Vmin will either diminish the potential leakage power saving or decrease yield. Another adaptive method is to use replica circuits. [59] propose to use replica cell which consists of two serial replicas of pull-up

transistor in parallel with two serial replicas of pull-down transistor to control the cell bias. To compensate V_T variation, multiple replica cells are implemented in parallel. However, it is inaccurate to simply assume that the maximum data retention voltage is $2V_T$ because the true data retention voltage is determined by the mismatch between the strength of the two inverters. In fact, when V_T is lower, larger cell bias is needed to retain data. One the contrary, when V_T is higher, smaller cell bias can retain data. Therefore, a scheme that can truly immune to all the process variation and environmental changes while maintaining the lowest acceptable yield is desired to achieve the maximum power reduction.

In this chapter, we propose a closed-loop V_{DD} scaling approach based on canary replicas, which allows aggressive power savings by tracking the impact of PVT variations on DRV. The remainder of the chapter is organized as follows. We first present the principle of the system. Then we describe the details of the major components, the adaptive reaction, and the overhead sources. Finally, we present the measurement results from the 90nm test chip.

4.2 Canary Adaptive System

I propose a feedback system that can lower V_{DD} for aggressive leakage power savings while protecting data by keeping V_{DD} above the data retention voltage (DRV) of the SRAM cells. Figure 4.2(a) shows the basic architecture of the system. A voltage regulator supplies V_{DD} to the core cells and to the canary replicas. When entering the standby mode, the controller starts lowering V_{DD}. Several banks of canary cells are designed to fail across a range of voltages above the actual DRV of the tail of the core bitcells. Canary cell failures are monitored by the online failure detector. If a failure is detected, then the controller raises



(a) (b)

Figure 4.2: (a)Architecture of the canary-based feedback loop for SRAM standby V_{DD} scaling and (b) Mechanism of the canary scheme.

V_{DD} to the last working value, resets the failed canary cells, and continues to monitor.

The big advantage of this feedback scheme will be the improvement of power savings. The online monitoring provided by the canary cells can track any global variations or environmental changes because they are affected by these changes in the same ways that the core cells are affected. Under PVT variations, the failure voltage of the canary replica changes by the same amount as the typical SRAM cell. Using feedback based on the canary failures, V_{DD} can be adjusted to approach the real SRAM failure point for the current scenario. Therefore, we can effectively remove the need to guard for the worst-case scenario and achieve more power savings for non-worst-case scenarios.

In addition, we have proposed a canary cell bank structure with a programmable failure threshold for trading-off SRAM data reliability with power savings. Figure. 4.2(b) illustrates the basic mechanism that provides this tradeoff. The canary cell bank contains multiple (n) canary sets, which fail at a regular intervals above the average DRV of the core

cells and maintain this behavior despite changes in global variations and environmental conditions so that V_{DD} can adjust with those changes. Local variation smears the distribution of canary DRVs in each set, but the canary distributions are not good indicators of the core cell distribution because there are too few canary cells. We will emphasize that the purpose of the canary categories is not to estimate the full distribution of the core SRAM cells, but instead to sense the proximity of the currently applied V_{DD} to the DRV of the average SRAM cell. To assess the proximity of failure to the tail of the distribution, we must model or measure that tail and relate its location to the canary behavior, as we will discuss in Section III. Providing a continuum of canary failures at voltages above the DRV of the average core bitcell allows the designer to set and to alter the tradeoff between storage reliability and power. This architecture allows for a variety of power-saving policies, and we provide a simple one as an example. Consider a handheld device holding video data during standby. When power saving is the major concern and losing a few bits of this data is acceptable (e.g. when using an ECC method), a failure threshold may be quite near (or below) the real tail of SRAM array-wide DRV. When the application changes and data are more important, the failure threshold can be reset to a higher value. This makes the controller raise V_{DD} until meeting the new failure threshold to provide a larger margin of protection above the array-wide DRV.

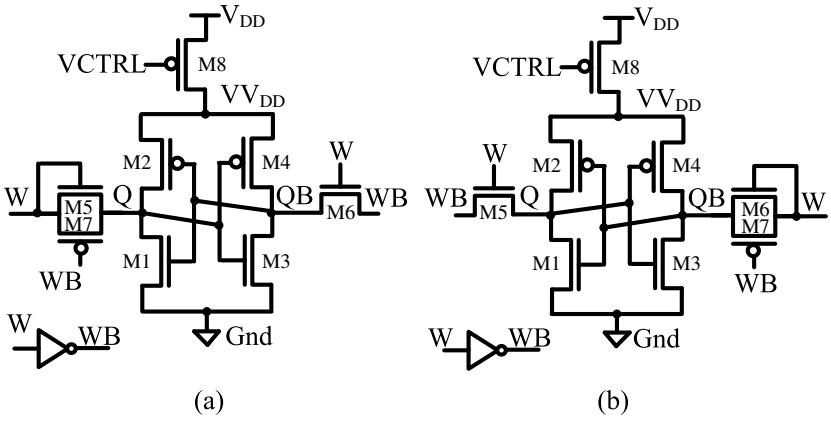


Figure 4.3: (a) Canary cell ‘1’ schematic. (b) Canary cell ‘0’ schematic.

4.3 Major Components

4.3.1 Canary Cell and Canary Bank

The canary cell is the most important component in our system. It must duplicate the impact of global changes on the core SRAM cell stability. Also, the canary cell must fail before the SRAM cells to prevent the loss of data in SRAM. Although local variation widens the distribution of canary DRVs, the canary distribution is not a good indicator of the SRAM cell distribution because there are too few canary cells. Therefore, we must use a design that makes it more sensitive to V_{DD} than it would be simply due to the impact of local variation. We proposed the circuit in Figure 4.3(a) and (b) as canary cells to hold ‘1’ and ‘0’, respectively [63]. The canary cell ‘0’/‘1’ contains the same 6T transistors (M1∼M6) as any SRAM cell. Q and QB are the internal storage nodes. To enhance the write capability at sub-threshold supply voltages (e.g. for canary reset), another PMOS pass transistor (M7) is added to the side of the cell that stores a ‘1’. The input signal, W,

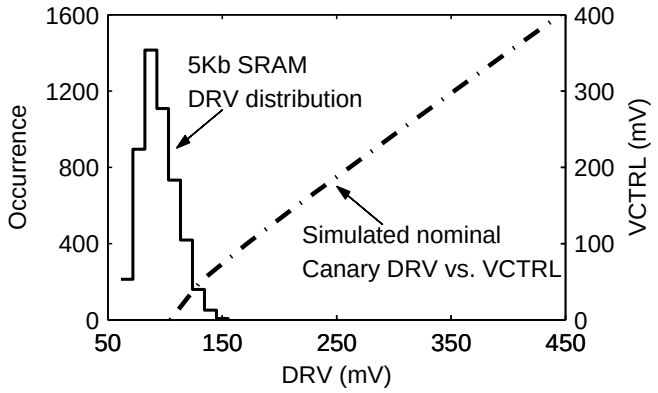


Figure 4.4: Simulated nominal canary cell DRV vs. VCTRL relative to a 5-Kb SRAM DRV distribution.

and its inversion, WB, act as the bitlines and wordline for writing data to the cells during a reset. When W is high, the canary cell resets its data; when it is low, the canary cell enters the standby mode. A PMOS header (M8) is inserted between the supply voltage of the canary cell and V_{DD}, and another input signal VCTRL drives its gate.

These small circuit modifications contribute to a higher DRV (failure voltage) for canary cell. First, the reset circuit that writes the cell applies the worstcase vector to the bitlines during hold. For example, the canary cell ‘1 (Figure 4.3(a)) has nodes Q and WB at ‘1 while QB and W remain ‘0 during standby. This creates the worst-case leakage through the access transistors to encourage the cell to flip, increasing the mean of DRV by 10.65%. A more efficient way to change the canary DRV is to tune the strength of the PMOS header. A larger VCTRL value increases the resistance of the header, and the actual supply voltage of the canary cell, VV_{DD}, drops lower than V_{DD}. This powerful knob essentially moves the mean of the DRV distribution for each canary cell across a wide range (as desired in Figure 4.2(b)). Figure 4.4 shows the simulated DRV of the nominal canary cell vs. VCTRL relative to the core cell DRV distribution. It is clear that the control of the header

allows us to provide the desired continuum of failure voltages for the canary cells. It also illustrates the approximately linear relationship between the canary DRV and VCTRL, so canary DRVs can be placed at regular intervals above the core DRV using evenly spaced VCTRLs.

Usually one cell can store either state '0' and state '1'. Due to device mismatch, one state becomes more stable than the other. But it is uncertain which state is more stable because of some random sources of device mismatch. Thus it is possible that the canary cell happens to be written with the more stable state and then will never flip. To solve this data dependency, we first simply use the redundancy technique. We group one canary cell '0' and one canary cell '1' into one canary set, and perform the OR operation to set the failure status of the canary set when either canary cell '0' or canary cell '1' fails. This redundancy approach is easy to implement, but there is the drawback limiting its efficiency, which we will discuss more in Chapter 5.1.

Different canary categories are organized in the bank structure as shown in Figure 4.5. The canary bank contains multiple sets (rows) of canary cells (e.g. 1-cell/row), and each set shares a distinct VCTRL values. A programmable failure threshold allows a range of policies for trading off power and reliability. We employ 3-way redundancy of the banks with majority-3 gates to screen out abnormalities caused by rogue cells with large variation. The VCTRL values are set off-chip or by an on-chip resistor ladder that generates evenly spaced VCTRL values between the voltage rails.

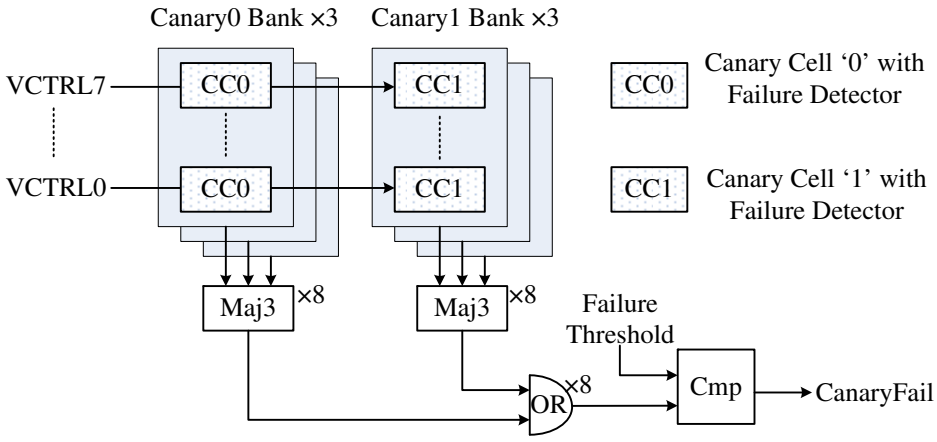
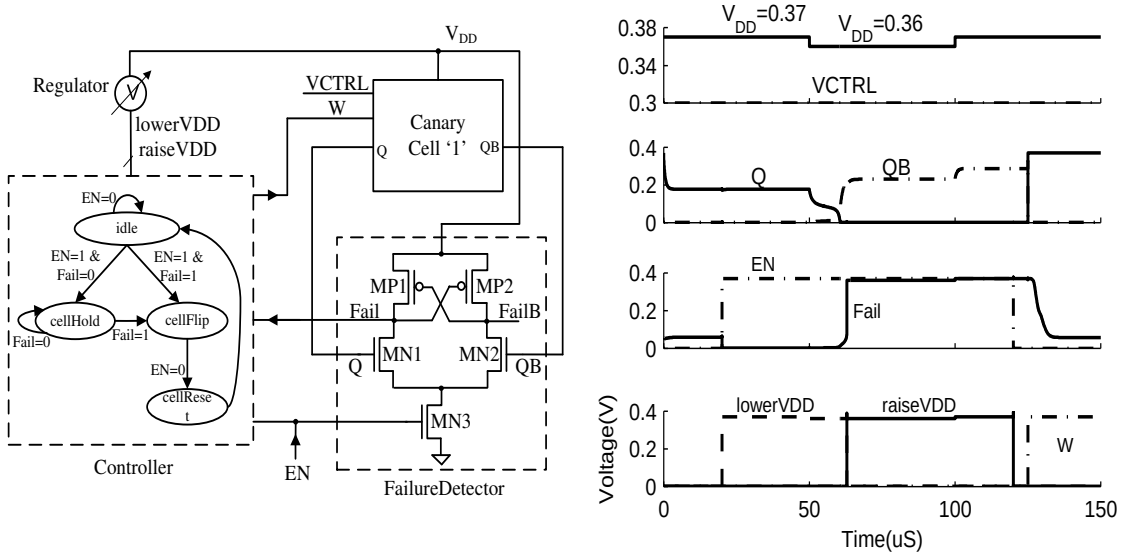


Figure 4.5: Canary bank structure.

4.3.2 Failure Detector and Canary Controller

Without a robust failure detector and a good controller, the feedback system can not successfully adjust V_{DD} even if the canary cells properly send out a failure alarm. Therefore, both the failure detector and the controller are critical components for our feedback system.

Figure 4.6(a) shows the proposed circuit and structure for these components. Each canary cell connects to its own failure detector through the storage nodes Q and QB. Once Q and QB flip or converge to a single value, the detector should be able to capture that and assert the output 'Fail' signal. Since the flipping failure is the major concern for the canary cell due to the asymmetrical bitlines, we propose a static sense-amplifier as the failure detector. It shares V_{DD} with the canary cell. The inputs to the differential pair MN1 and MN2 come directly from the canary cell. For the canary cell '1' (Figure 4.3(a)), Q connects to MN1 in this example; for the canary cell '0' (Figure 4.3(b)), QB should connect to MN1 instead.



(a) (b)

Figure 4.6: (a) Canary failure detector and controller. (b) Timing diagram of detecting failure and resetting canary cell '1' when VCTRL is 0.3V. The DRV of the canary cell is 0.37V.

For simple illustration, only a single canary cell and its failure detector is shown here to connect with the controller. In fact, the failure signals from all the canary cells will be processed in order to generate a final 'Fail' signal for the controller. Since we actually employ 3-way redundancy in the banks of canaries in our test chip to reduce the impact of local variation on canary cells, the failure signals from the redundancy banks first go through the majority-3 gates to screen out abnormalities caused by rogue cells with large variations. Then the failure signals from canary bank '0' and canary bank '1' combine through the OR gates before comparing with the failure threshold. The failure threshold is a 8-bit value preloaded before operation. If the generated failures are larger than the failure threshold, a final failure signal will be asserted and sent to the controller. This is the signal that causes the controller to raise V_{DD} slightly and reset the canary cells.

Figure 4.6(a) also illustrates the state transistions in the controller that we implemented on the 90nm test chip. There are four states: idle, cellHold, cellFlip and cellReset. The controller receives 'Fail' signal from the failure detector, and sends out the two control signals 'lowerVDD' and 'raiseVDD' to the regulator and 'W' signal to the canary cell for resetting.

Figure 4.6(b) gives the timing diagram that shows how the states transfer for canary cell '1' with a 0.3V VCTRL value. When V_{DD} is 0.37V, Q and QB hold their original value. After we assert the enable signal 'EN', the failure detector evaluates Q and QB, and then 'Fail' goes to zero, which makes the controller change from the 'idle' state to the 'cellHold' state, and the signal 'lowerVDD' rises up to inform the voltage regulator to decrease V_{DD} by one step of 0.01V (for example). Once V_{DD} is lowered to 0.36V, Q and QB flip to the opposite value, and hence 'Fail' rises up and the 'cellFlip' state becomes valid. This state asserts 'raiseVDD' to make the regulator increase V_{DD} by one step and thus go back to the previous value 0.37V, which is actually the DRV of this canary cell. After V_{DD} has raised up to 0.37V, 'EN' goes low to disable the failure detector and the controller enters the 'cellReset' state, which asserts the 'W' signal to write the original values into Q and QB.

Both the failure detector and the contoller have been implimented in our test chip and measured to function correctly at low V_{DD}.

4.4 Adaptive Reaction to Environment

In this section, we extend our previous work by taking a detailed look at the ability of the canary cells to track the impact of global effects on the core SRAM cells. We present a

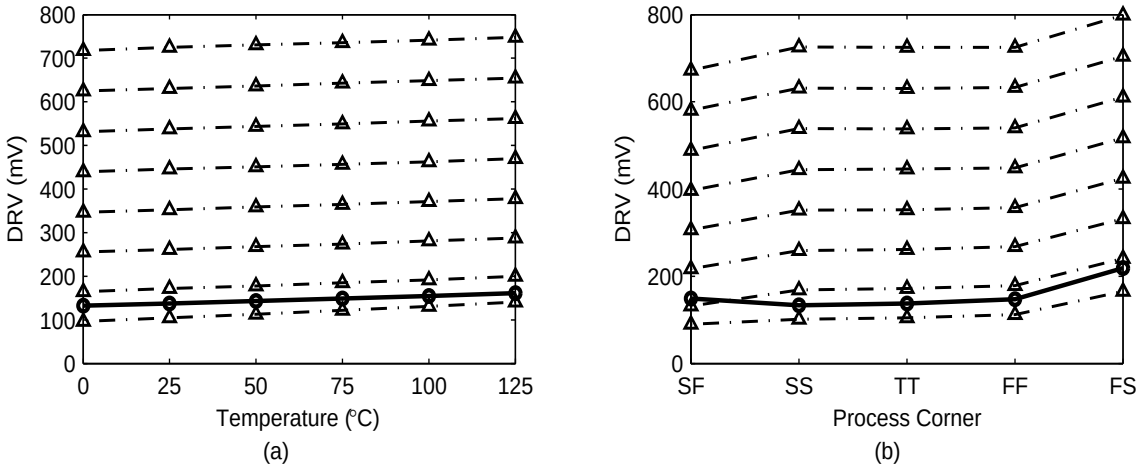


Figure 4.7: Simulated DRV of the canary sets (lines with triangles and the upper ones have higher VCTRL) and the worst DRV of a 1-Kb SRAM (the line with circles) change consistantly with (a) temperature and (b) process corner for the 90nm technology.

new analytical model that maps the VCTRL voltage to the DRV of the canary cells.

4.4.1 PVT Variation Tracking

One of the most important traits of the canary cell is its ability to track global PVT effects on the core SRAM cells. Without this characteristic, the feedback system cannot react properly under global stimuli. So it is necessary to examine the canaries under different PVT variations.

Figure 4.7(a) and (b) show simulated results that compare the canary behavior with an SRAM array across temperature changes and global process corners, respectively. We used a 1-Kb SRAM as an example. In this figure, the curve with circles stands for the worst DRV of the 1-Kb SRAM, and the curves with triangles stand for different canary sets (the upper ones are the sets with higher VCTRL). The upper 7 canary sets consistently fail before the SRAM at all the temperatures and all the process corners with the only exception

of the SF (Slow-N Fast-P) corner. This indicates that the canaries will successfully track global effects on the SRAM array. The one exception occurs because our technology is a strong-NMOS process (e.g. NMOS is noticably stronger than PMOS in sub-threshold). At the SF corner, the impact of the global process variation becomes too weak compared with the local variations, so the whole SRAM DRV distribution (or tail) is not strongly influenced by this process variation for the canary set designed to fail closest to the core cells. If temperature gradients are a concern, then canary cells can be dispersed at different locations in a core array. If voltage fluctuation occurs, the DRV of core cells and canary cells will increase/decrease by the same amount because they are sharing the same power supply. Therefore, the canary cells are able to track PVT variations.

4.4.2 Models for Adaptive Setting

Previously we have mentioned that our feedback scheme has the flexibility to trade off SRAM reliability with leakage power savings. This ability is dependent on the appropriate setting of the canary cells and failure threshold for a given SRAM for a required constraint on either relibility or power consumption. To make these settings more quickly and precisely, here we present two models to estimate SRAM DRV and canary DRV, and we integrate them into a final model that can directly compute the necessary canary VCTRL values to provide a desired level of SRAM reliability.

We have previously proposed the CDF and inverse CDF models for an SRAM DRV distribution in [65]. Equation (4.1) is the inverse CDF model of DRV:

$$ {F_{DRV}}^{-1}(x) = \frac{1}{k}\left[\sqrt{2}\sigma_0 \cdot erfc^{-1}\left(2 - 2\sqrt{x}\right) - \mu_0\right] + V_0. \tag{4.1} $$

where x is the probability that $DRV < {F_{DRV}}^{-1}(x)$, k is the slope of SNM High (SNM

for holding '0') versus V_{DD}, μ_0 and σ_0 are the mean and standard deviation of SNM High at $V_{DD} = V_0$, and $erfc^{-1}(\cdot)$ is the inverse complementary error function. k, μ_0 and σ_0 are fitting coefficients; k can be extracted from a DC sweep simulation and μ_0 and σ_0 can be extracted from a small-scale (1.5K-5K) Monte-Carlo simulation. This model has shown a high accuracy in comparison with Monte-Carlo simulation out to 6σ as well as in comparison with the Statistical Blockade tool [57] beyond 6σ, which uses a fast Monte-Carlo method to filter the tail samples and fit them to a Generalized Pareto Distribution (GPD) model.

In this paper, we present a new model to estimate the canary DRV. As observed in Figure 4.4, the canary DRV changes approximately linearly with VCTRL. This linear dependency can be modeled by analyzing the current through the PMOS header M8 (in Figure 4.3(a)). Let us assume the minimum current to hold the canary cell data is I_{min}, which occurs when the actual supply voltage of the canary cell, VV_{DD}, is equal to the cell DRV. Because the cell operates in the sub-threshold region during the data retention mode, M8 will also operate in the sub-threshold region. So the leakage current through M8, I_8, is

$$I_8 = I_0 \cdot exp\left[\frac{V_{DD} - VCTRL - V_{T8} + \eta_8(V_{DD} - DRV)}{n_8 V_{th}}\right]\left[1 - exp\left(\frac{-V_{DD} + DRV}{V_{th}}\right)\right]. \tag{4.2}$$

where V_{T8} is M8's threshold voltage, η_8 is its DIBL coefficient, n_8 is its subthreshold swing factor, V_{th} is the thermal voltage, and I_0 is the off current. I_8 is also equal to $I_2 + I_4$, where I_2/I_4 is the leakage current through M2/M4. For a given canary cell, we assume that the DRV remains the same no matter what VCTRL is. This is reasonable because M1$\sim$M7 are not changed. Therefore, I_2 and I_4 keep constant and so does I_8. If we define that constant

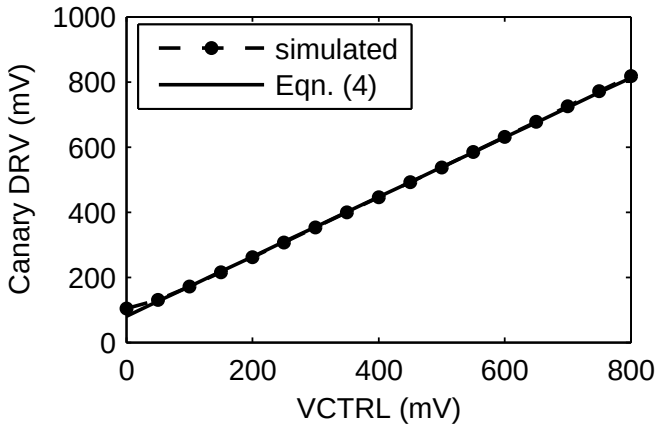


Figure 4.8: Estimated canary DRV from (4) vs. VCTRL compared with the simulated results.

as I_C, (4.2) can be written as

$$exp\left[\frac{(1+\eta_8)V_{DD}-VCTRL}{n_8V_{th}}\right]\left[1-exp\left(\frac{-V_{DD}+DRV}{V_{th}}\right)\right]=\frac{I_C}{I_0}exp\left(\frac{V_{T8}+\eta_8 DRV}{n_8V_{th}}\right). \tag{4.3}$$

Since the right hand of (4.3) are all constant values, we can simply replace them with a constant K. Furthermore, when V_{DD} is much larger than the DRV, we can ignore the roll-off term. So finally, we can derive that:

$$V_{DD}=\frac{VCTRL+n_8V_{th}ln(K)}{1+\eta_8}=\frac{VCTRL}{1+\eta_8}+b. \tag{4.4}$$

which verifies the linear relationship between the canary DRV and VCTRL and implies that the slope is about $1/(1+\eta_8)$. With an initial pair of $VCTRL_0$ and V_{DD0}, we can obtain the offset value b. Figure 4.8 compares the canary DRV values from (4.4) with the simulated results. This first-order linear model provides a good approximation for most VCTRL values. However, when VCTRL is less than 100mV, the model is less accurate because V_{DD} is near the actual cell DRV, and the rolling-off term cannot be ignored, in which case (4.3) is a more accurate equation.

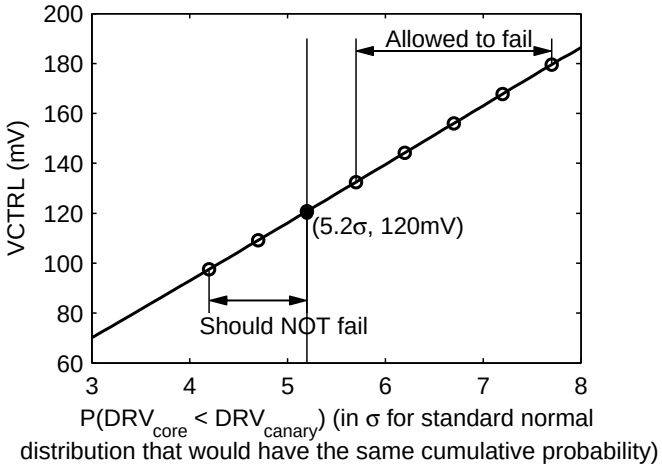


Figure 4.9: Estimated VCTRL value vs. the probability that DRV_{core}<DRV_{canary} (in σ). Failure threshold (the vertical line) is set according to the reliability constraint, e.g. 5.2σ. Only the canary sets on the right side of the failure threshold (the upper 5 sets here) are allowed to fail.

Now combining (4.1) and (4.4), we can estimate the VCTRL value necessary to satisfy a given SRAM reliability constraint with (4.5):

$$VCTRL = \frac{1+\eta_8}{k}\left[\sqrt{2}\sigma_0 \cdot erfc^{-1}\left(2-2\sqrt{x}\right)-\mu_0\right]+(V_0-b)\cdot(1+\eta_8). \qquad (4.5)$$

where x is the probability that SRAM DRV (DRV_{core}) is less than the canary DRV (DRV_{canary}) with the desired VCTRL value, and all the other parameters are the same as in (4.1) and (4.4). Figure 4.9 shows the estimated VCTRL values from (4.5) with the solid curve. In this figure, the probability is expressed by σ, which is the equivalent point for a standard normal distribution that would have the same cumulative probability. For example, if 5.2σ probability is required (for a fault-free 10-Mb SRAM), VCTRL is about 120mV. This implies that canary cells with VCTRL larger than 120mV have an even higher probability of failing before all of the core cells. Now with the required SRAM reliability constraint, we can set an appropriate canary failure threshold. As in the example in Figure 4.9, if at least

5.2σ reliability is needed, we can consider the vertical line at 5.2σ as the failure threshold and select the point (5.2σ, 120mV) as one of the canary sets. Then we can pick the other 7 points along the solid curve for the remaining canary sets so that the entire possible SRAM data reliability range can be covered. Here, we selected 5 points with VCTRL higher than 120mV and 2 lower ones as an example. This configuration means that the feedback loop will allow only the upper 5 rows of canary sets (corresponding to the upper 5 points) to fail. We can know the approximate reliability of the core SRAM cells by observing the failure status of the canary sets. If the application changes and needs a higher reliability, we can reset the failure threshold for the current canary configuration or even reconfigure all of the canary sets (by remapping VCTRLs) for better results.

4.5 Overhead Analysis

Thus far, we have analyzed the benefits of using canary-based V_{DD} scaling without accounting for overhead. In this section, we quantify the impact sources of overhead on the potential savings achievable by our scheme.

4.5.1 Canary Circuit Overhead

Our test chip has shown only about 0.6% area overhead due to the canary circuits. We can expect even smaller area overhead for systems with larger memory blocks.

The canary power overhead includes the power of canary array (48-b canary cell) and peripheral (including all the failure detectors, controller, and other components) circuits. The dynamic power of the canary circuits is small relative to their leakage power since

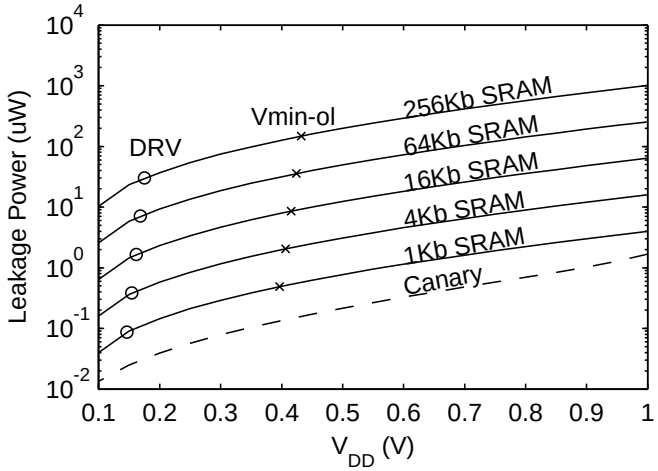


Figure 4.10: Leakage power consumption of SRAM array with different size as well as canary power overhead at typical PVT scenario.

the canary system works at a very low frequency. We can thus consider the total overhead power to equal the leakage power of the canary circuits. Figure 4.10 shows the leakage power of differently sized SRAM macros as well as the simulated canary circuit leakage power for the average (typical) PVT scenario. To account for local variation impact on the SRAM array, Monte-Carlo simulation with mismatch can be used. However, for big arrays, running M-C simulation is too expensive. We use an alternative fast way that obtains SRAM leakage power from the statistic of the cell leakage current. We get I_{cell}, the mean of the cell leakage current distribution, from a 5000-iteration Monte-Carlo simulation. The leakage current of a N-bit memory can be approximated as a normal random variable with the mean of NI_{cell} by applying the Central Limit Theorem. In Figure 4.10, each solid curve denotes the average leakage power of the corresponding SRAM macro. The circle point on the curve denotes the DRV tail of this SRAM macro at typical PVT scenario, while the cross point denotes the open-loop Vmin 'Vmin-ol' (i.e. V_{DD} when the worst-case PVT scenario has a 50mV SNM margin) for this SRAM macro. Both the DRV and Vmin-ol

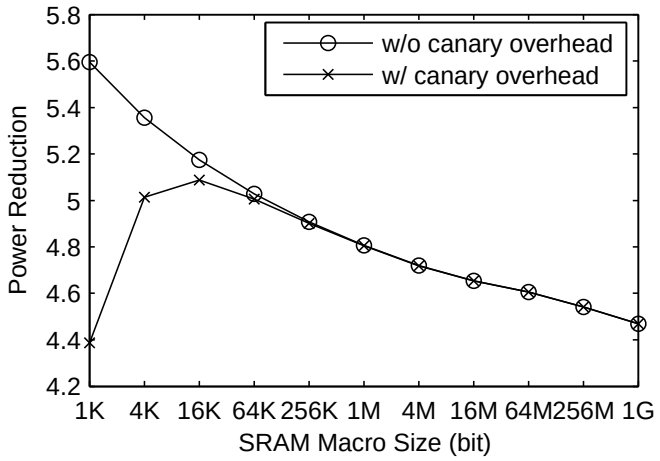


Figure 4.11: Power reduction of using canary approach relative to the open-loop approach vs. SRAM size (with or without taking account of the canary overhead) at typical PVT scenario.

values are obtained from our DRV inverse CDF model (4.1). With an SRAM larger than 64-Kb, the power overhead of all of the canary circuits is at least 2 orders of magnitude smaller than SRAM leakage power, so it is negligible.

Figure 4.11 shows the power reduction achieved by the canary approach relative to the open-loop approach for different SRAM macro sizes for the typical PVT scenario. For each SRAM size, the canary approach uses the DRV value in Figure 4.10 as the standby V_{DD}. While the open-loop approach uses the Vmin-ol value in Figure 4.10 as the standby V_{DD}. Without considering the canary overhead, the smallest SRAM has the largest saving because it is possible to lower V_{DD} farther using feedback due to the lower DRV. However, accounting for the canary overhead shows that the effective savings with the smallest SRAM are reduced. It should be noted that all the SRAMs up to 1-Gb have more than $4\times$ of power reduction compared with the open-loop approach. It indicates that our canary scheme is efficient for any size of SRAM in terms of leakage power reduction, even when accounting for canary overhead power.

4.5.2 DC-DC Converter Overhead

Our canary-based V_{DD} scaling approach requires a DC-DC converter that can provide a standby supply voltage across a fairly large range of values. Since this low, variable voltage is only supplied during standby, the load current may be relatively low. The DC-DC regulator may be on-chip or off-chip, but either way, we need to account for the impact of its effeciency on the overall power savings from using our approach. [53] has described a switched DC-DC converter that can deliver load voltages ranging from 0.3V to 1.1V. That particular converter provided $>$70% efficiency over a wide range above 0.45V. The minimum efficiency remained larger than 55%. This converter shows the sort of efficiencies that we might expect to see in this space. The recent interest in low voltage operation is leading to further investigation of regulators with higher efficiencies that are tailored specifically to low supply voltages.

Figure 4.12 shows the leakage power reduction factor for a 1-Kb SRAM by using our canary approach relative to using the nominal V_{DD} when accounting for a range of DC-DC converter efficiencies. The power reduction achieved by using the open-loop V_{DD} scaling approach is also shown. Best-case (b-c), typical (typ) and worst-case (w-c) PVT global scenarios for each approach are simulated. The open-loop approach uses the Vmin-ol value as shown in Figure 4.10 so that the worst PVT scenario can have a guard-band of 50mV SNM margin. The canary approach can apply V_{DD} down to the actual DRV of the given PVT global corner without losing data, so additional power savings can be achieved. However, when accounting for a non-ideal DC-DC efficiency, the actual power of both the canary approach and the open-loop approach will increase due to the overhead of the DC-DC converter. Under a given DC-DC efficiency, the power reduction factor

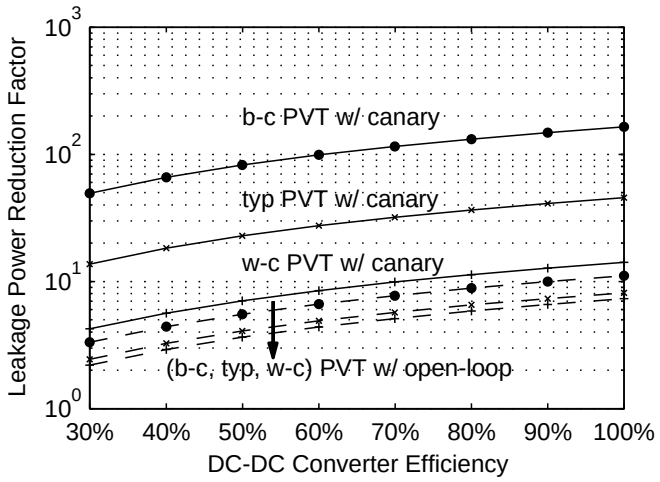


Figure 4.12: Power reduction of 1-Kb SRAM using canary or open-loop V_{DD} scaling when DC-DC converter efficiency is considered. Power reduction is relative to the power consumed at the nominal V_{DD} (1.0V). Best-case (b-c), typical (typ), and worst-case (w-c) PVT scenarios for each approach are shown.

is the ratio between the power consumed without V_{DD} scaling (i.e. V_{DD}=1.0V) and the actual power consumed with V_{DD} scaling. When the converter efficiency is only 30%, the canary approach gains $\sim$50$\times$ and $\sim$14$\times$ of power reduction while the open-loop approach gains $\sim$3$\times$ and $\sim$2$\times$ of reduction at the best-case and typical PVT scenario, respectively. Therefore, even when using a non-ideal DC-DC converter, it is obvious that V_{DD} scaling (by either open-loop or our canary approach) can bring substantial power reduction. In addition, for both the typical and the best-case PVT scenario, the power reduction from the canary approach with 30% converter efficiency is still higher than that from the open-loop approach with an ideal converter (100% efficiency), which demonstrates that our canary approach can effectively achieve extra power savings over the open-loop approach even under the condition of using a less-efficient converter.

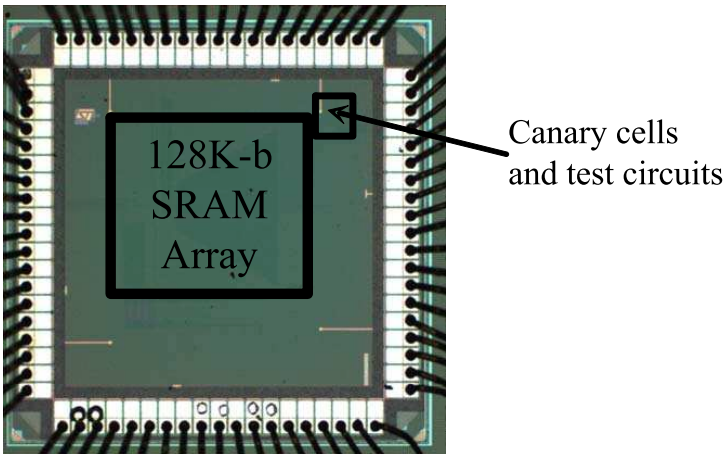


Figure 4.13: 90nm chip die photograph.

4.6 90nm Test Chip Implementation and Measurement

A 90nm CMOS bulk test chip implements all of the circuits that we have described except the V_{DD} regulator [63]. Figure 4.13 is the die photo of the chip. Figure 4.14(a) shows the measured average DRV of canary cells versus VCTRL at room temperature. The canary cells exhibit the desired linear dependency on VCTRL. We also measured canary DRV with VCTRL at different temperatures as shown in Figure 4.14(b), which demonstrates that the canary cells successfully track temperature changes. In this paper, we will present additional measured results from the 90nm test chip. Figure 4.15 is the measured DRV histogram of one 8-Kb SRAM array and the measured DRV histogram of 5 canary categories (with VCTRL values ranging from 0mV to 800mV with a step of 200mV). For testing the DRV distribution of one canary category (e.g. Canary #3), we use a test mode that sets all of the canary cells on the test chip to have the same VCTRL value (e.g. 400mV) supplied by an external reference source. The measured SRAM array DRV values range

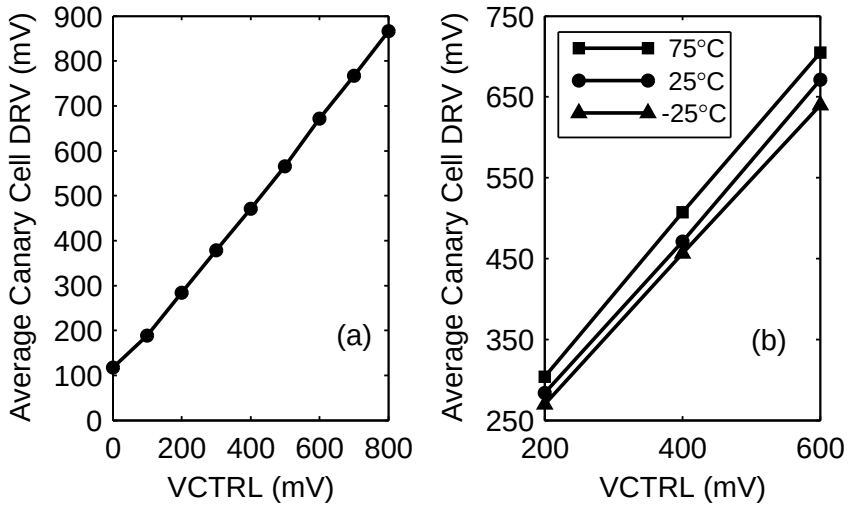


Figure 4.14: Measured average canary DRV vs. VCTRL at (a) room temperature and at (b) different temperatures.

from 60mV to 350mV with a mean value of 112mV and a standard deviation of 22mV. This wide distribution confirms the expected effects of local mismatch in the chip. Each measured canary category has a relatively narrower spread compared with the SRAM cells, and each one has a similar distribution. By applying different VCTRL values, we locate the failure voltage of different canary categories across a broad range that starts within the failure range of the core SRAM cells and extends to voltages well above the failure range of the core. This measured result proves the feasibility of implementing the tradeoff between SRAM reliability and leakage power, which was illustrated in Figure 4.2(b).

We also tested one closed-loop control method shown in Figure 4.16. A VCTRL generator (implemented as a resistor ladder) shares V_{DD} with the SRAM core cells as well as the canary cells. It consist of 8 indentical resistors so that 8 regularly spaced voltage reference values can be generated. These nodes serve as the VCTRL signals and connect to the corresponding canary category (set). Hence in this test mode, the canary sets would fail in a

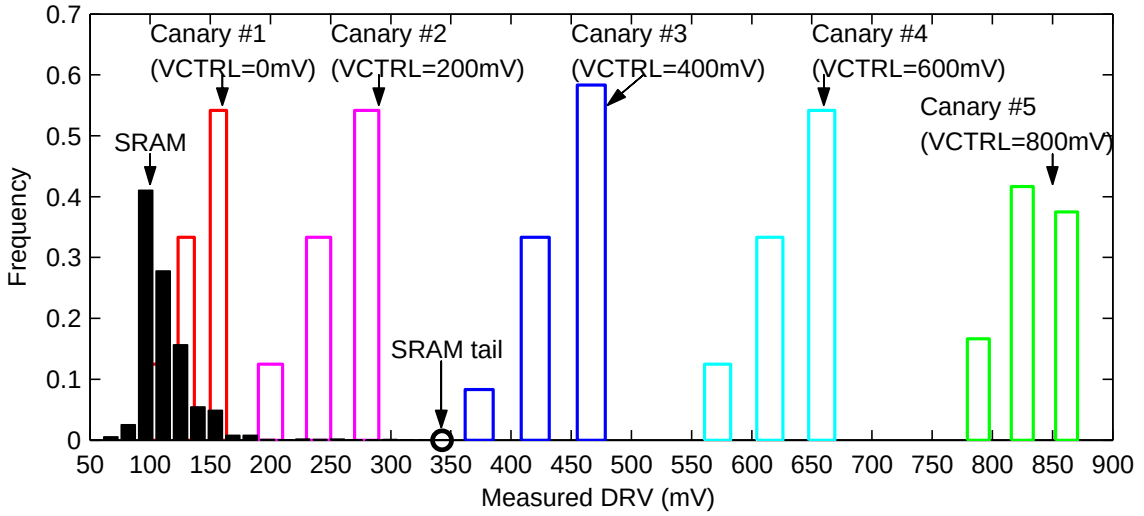


Figure 4.15: Measured DRV histogram of one 8-Kb SRAM array and measured DRV histogram of 5 canary categories. The circle denotes the tail of the measured SRAM DRV distribution.

sequence from set #7 to #0 as we continuously scale V_{DD}. Figure 4.17 shows the measured results using this method. Here, each column is one canary set and each row shows the status of all the canary sets at one V_{DD} point. The cross symbol means the canary set fails and the circle symbol means it holds its data. For example, at V_{DD}=0.6V, the upper 3 canary sets (with higher VCTRL) have failed, but the lower 5 sets continue to successfully hold their data. When further reducing V_{DD} to 0.5V, the canary set #4 fails while the lower 4 sets keep holding their data. This figure demonstrates that lowering V_{DD} encourages more canary cells to fail, which then implies closer proximity to the failure of the core SRAM cells.

The measured leakage power of the SRAM array on one die with V_{DD} scaling is shown in Figure 4.18. Without losing generality, we can assume 0.7V to be the standby V_{DD} for the worst-case scenario among all the PVT variations. Then for the die with the DRV tail at 0.35V under normal environmental conditions, our canary-based feedback approach

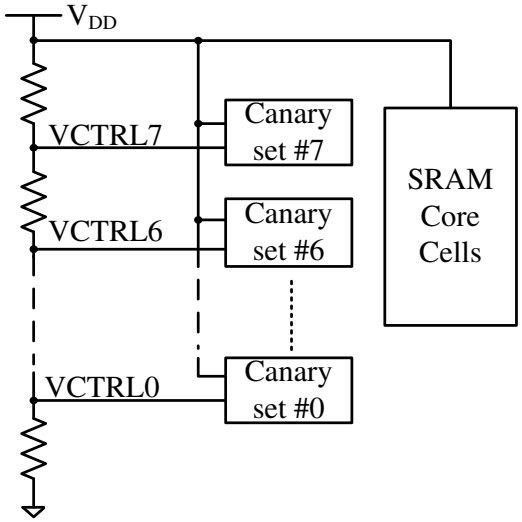


Figure 4.16: One closed-loop measurement sturcutre.

can adjust V_{DD} to this value and thus bring $\sim 5\times$ more power savings compared with the conservative worst-case-based approach, and $\sim 11\times$ compared with using the nominal V_{DD}.

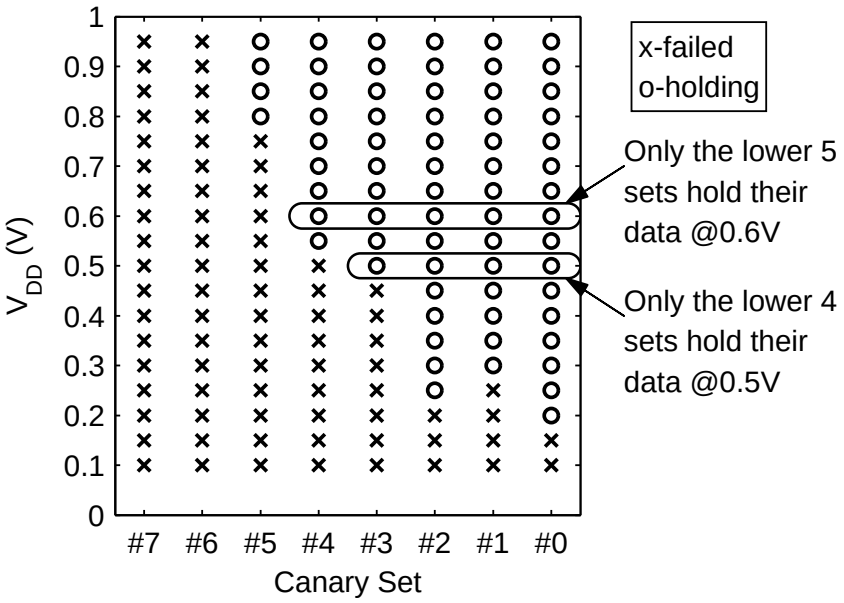


Figure 4.17: Measured failure status of each canary set with V_{DD} scaling.

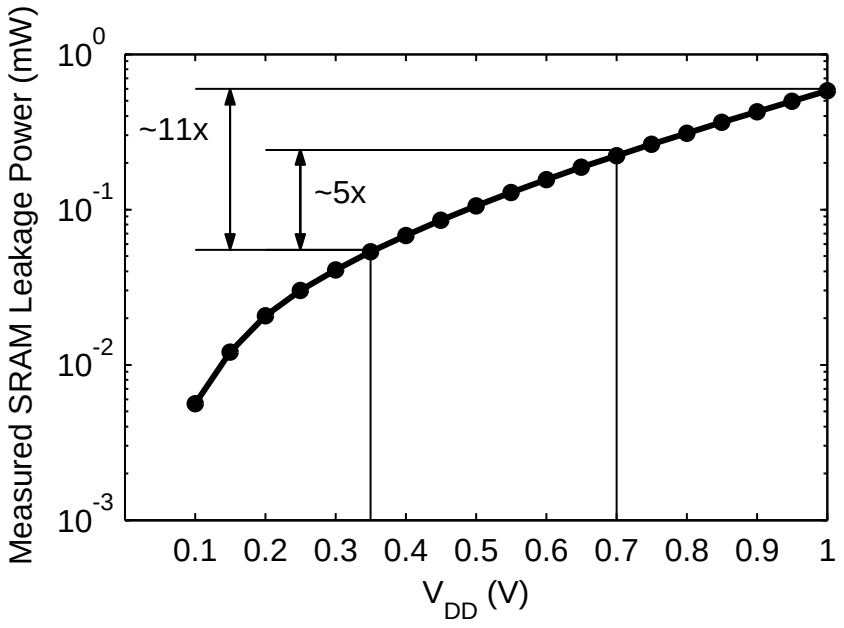


Figure 4.18: Measured 128-Kb SRAM leakage power vs. V_{DD}.

Chapter 5

Enhanced Canary System for 45nm and Beyond

In this paper, we propose several enhancements for variation adaptation and self calibration. To improve the canary cell, we propose to add dummy bitcells around the canary cell so that it behaves more like an SRAM cell in the presence of variation. We also propose a new canary circuit which always tunes the cell to its less-stable state to avoid the possibility that the canary cell would never fail because it happens to hold its more-stable value. We incorporate a builtin self-test (BIST) block to automate the calibration of the worst SRAM data retention voltage (DRV) and the tuning of the initial failure threshold for adapting process variation. To further demonstrate the effectiveness of the canary system for SRAMs in sub-45nm nodes, we implement the canary system on a 45nm bulk test chip. The measurement results indicate that the addition of the dummy cells inside the canary cell can effectively reduce the variation of the canary cell upon itself and thus improve the accuracy of the tracking behavior. Finally, we show simulation results with predictive

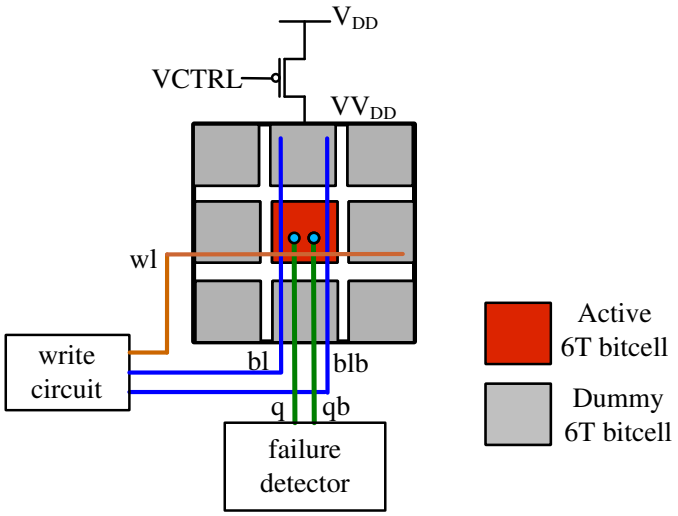


Figure 5.1: new canary cell structure with dummy cells; Only modification to active 6T layout is connecting to Q and QB.

technology models beyond 45nm.

5.1 Canary Cell Improvement

5.1.1 New Canary Cell Structure

We first propose a new structure for the canary cell to improve the correlation of global effects on canary cells and SRAM cells. Figure 5.1 shows the proposed strucutre. Around the actual functional bitcell, dummy bitcells are added to mimic the real physical environment of an SRAM cell. To reduce area cost, we use a 3×3 SRAM mini-array for each canary. A failure detector monitors the active bitcell in the center. To ensure the canary cell behaves more like SRAM cells in the presence of the global variations, we use the same layout pattern as the real SRAM array except some minor changes on metal wires for

pulling out the storage nodes of the central cell. The actual power supply of the mini array (VV_{DD}) is connected with the PMOS header. As before, when we tune VCTRL to a higher value, the PMOS header is partially turned on. Thus, to make the canary cell fail, a higher V_{DD} value must be applied so that the voltage of the VV_{DD} node can be smaller than the cell's actual data retention voltage.

5.1.2 New Circuit for Canary Cell Reset

Issue

As shown in Figure 4.5, we previously combined two separate canary cells, one for storing '0' ('cancell-0') and the other for storing '1' ('cancell-1'), within one canary set to account for the dependence of DRV on data pattern. Although this way is simple and easy to implement, it has one drawback. Mismatch causes a cell to be much more stable at one data value than the other, and it is uncertain which data value is more stable due to randomness of local variation such as from dopant fluctuation. For one canary category, if both 'cancell-0' and 'cancell-1' happen to be more stable at the value that they are holding, this canary category will never fail or fail at a very low supply voltage regardless of the VCTRL value.

This can be better explained with the help of DRV. We denote DRV0 as the DRV for holding a '0' and DRV1 as the DRV for holding a '1'. Figure 5.2(a) shows the correlation between DRV0 and DRV1 when both come from the same cell. 100 Monte-Carlo points are simulated. Majority of samples have one DRV value near or equal to 0 and the other much greater than 0 because device mismatch causes the cell to be unbalanced. Few samples have two values close to each other because they are more balanced. Note that DRV0

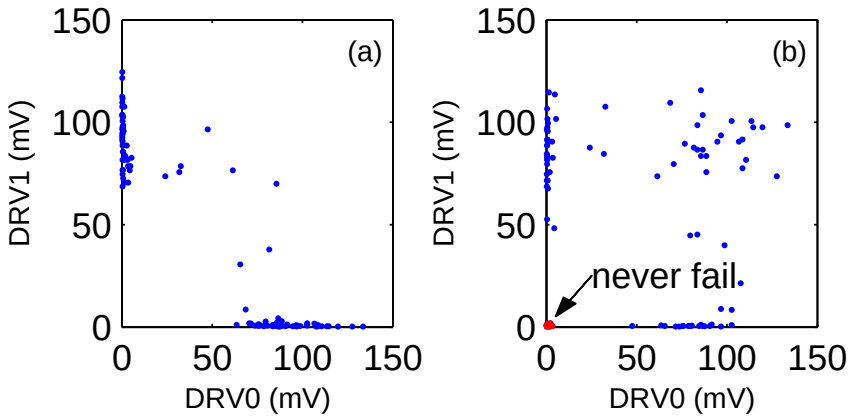
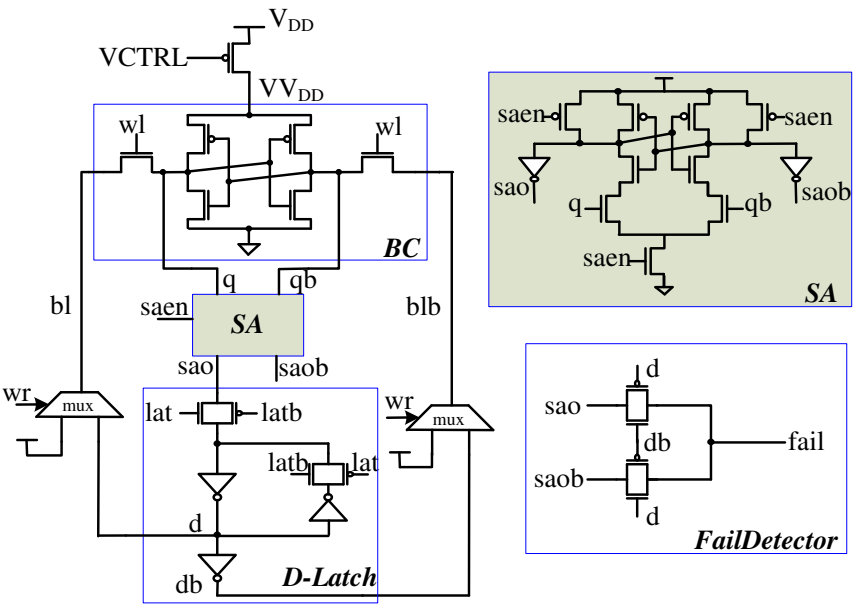


Figure 5.2: The correlation between DRV0 and DRV1 (a) when they come from the same cell and (b) when the come from separate cells. 100 samples are shown.

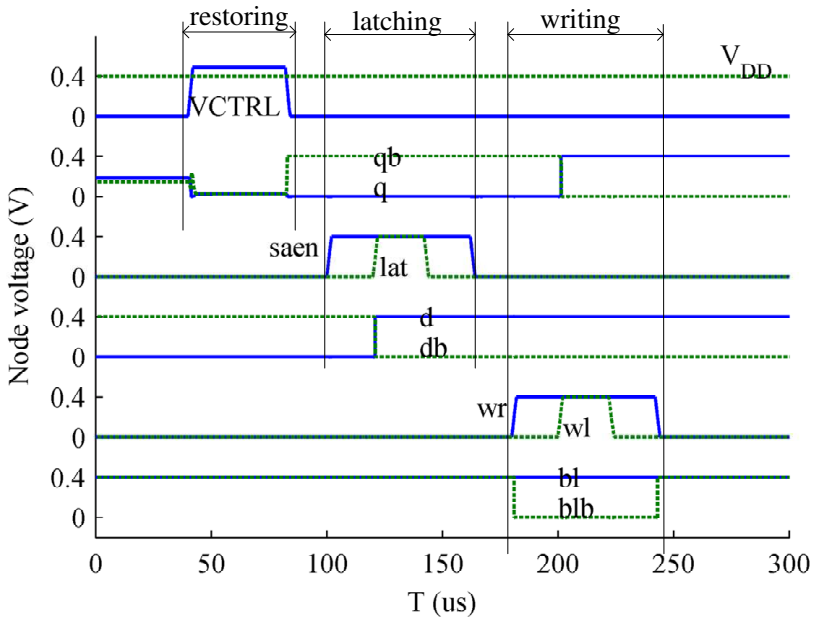
and DRV1 never simultaneously equals to 0. Now if the DRV1 comes from a separate cell, the correlation map between DRV0 and DRV1 changes to Figure 5.2(b). $\sim$20% of samples have both DRV0 and DRV1 near or equal to 0, which means both cells can always hold their respective data. We observed this issue in our test chip. Although we can use more redundancies of the canary sets to mitigate this issue, it degrades the accuracy of the tracking performance as well as the area efficiency.

Solution

To eliminate this issue, we propose a new circuit shown in Figure 5.3(a). Besides the mini-array in Figure 5.1 (simplified as a 6T bitcell for illustration here), the circuit includes a latching voltage-mode sense amplifier (SA), a D-latch, and two MUXs. Figure 5.3(b) shows the timing waveforms. There are three phases: the restoring, latching and writing phase. In the restoring phase, 'VCTRL' first rises to a high value. This turns off the PMOS header and leaves the actual power of the 6T cell (VV_{DD}) floating. The floating VV_{DD} has to be below the cell's DRV in order to reset the cell. So we use a boosted 'VCTRL'



(a)



(b)

Figure 5.3: (a) Circuit and (b) waveforms for canary cell self-loading less-stable state.

(higher than V_{DD}). After VCTRL returns back to 0, the storage nodes (q, qb) restore the cell's more-stable state (e.g. (0, 1) in Figure 5.3(b)). Then the latching phase starts with the rise of 'saen', which enables the SA so the stable state can be passed to the SA outputs (sao, saob). After some delay time, a pulse on the 'lat' signal allows the D-latch to capture the inverted 'sao' value into its output 'd'. So (d, db) are driven to the values of the cell's less-stable state (1, 0). 'saen' falls back to 0 at the end of the latching phase to disable SA. In the last writing phase, 'wr' rises first so that the MUX can select the value of (d,db) for the bitlines. Then a pulse of 'wl' writes the state (1, 0) into the canary cell, with which the canary cell could flip to the more-stable state at a relative higher DRV.

We also show the circuit for the failure detector in Figure 5.3(a). The failure detector performs XOR on (d,db) and (sao,saob). Once their values differ, it implies the cell has flipped and the 'fail' signal will be asserted. The restoring and latching phase only occur once for the system's entire operation (e.g. at start up). During the standby mode, when the canary failures disagree with the failure threshold value, the standby V_{DD} will be adjusted to a new value and then the writing phase will occur again to reset the cell with its less-stable state that is already stored in the D-latch. It should be noted that the supply voltage of the D-latch is directly connected with the V_{DD} of the SRAM cells. Less local variation occurs in D-latches with larger devices, so the D-latch can hold its data more reliably than SRAM cells during standby operation.

5.2 BIST and Tune

The benefit of our canary system is mainly contributed to the PVT-independent failure threshold, which determines the proximity of the applied V_{DD} to the tail of the SRAM DRV

as illustrated in Figure 4.2(b). The proximity implies the safety margin for V_{DD} scaling. No matter what PVT variation occurs, the predefined failure threshold ensures the same amount of proximity (i.e. safety margin) can be maintained. The only thing we need to do is to select an appropriate failure threshold according to the required power and reliability constraints before operation. Although the failure threshold can change with application requirements, there exists a boundary for the maximum power savings and minimum data stability. This boundary is actually determined by the closest proximity to the tail of the SRAM DRV distribution. However, the boundary varies from die to die because the DRV of SRAM cells as well as the DRV of the canary cells randomly spread due to the random sources such as the dopant fluctuation. Therefore, we must perform calibration to find the boundary of the failure threshold for each die. We propose to incorporate a built-in-self-test (BIST) block to automate this calibration process. The tail of the SRAM DRV distribution, Vmin, is first calibrated by the BIST. Then the boundary of the failure threshold can be found by applying that Vmin on the canary cells.

5.2.1 Calibrating SRAM DRV Tail

We first build a BIST to self-calibrate the tail of the SRAM DRV, i.e. the minimum supply voltage for SRAM (Vmin) during standby.

Authors in [21] also presented a calibration method for utilizing the source biasing approach. Source biasing raises cell source voltage (V_{SS}) to reduce SRAM leakage current. The highest V_{SS} value that can be applied to an SRAM is also limited by the worst cell stability. To find that value, they start with the nominal V_{SS} value (the lowest one, i.e. 0) and then increase V_{SS} until the SRAM can not tolerate more cell failures. For the V_{DD}

scaling approach, we could also start with the nominal V_{DD} value (the highest one) and gradually decrease V_{DD} to find the lowest V_{DD} value for SRAM data retention. However, with this searching method, a full scan of all the SRAM cells will be executed for each iteration until the number of the failures exceeds the number of errors that can be tolerated. For large SRAMs, one complete check of all the cells can cost a large amount of time. To save test time for big SRAMs, here we propose a faster searching method. Instead of the typical searching direction, we use the opposite direction, i.e. from lower V_{DD} values to higher ones. For one iteration, now we can stop checking the remaining cells once the number of failures exceeds the error tolerance limit. We can further accelerate the process by choosing the start point closer to the simulated average value of the cell DRV instead of 0. Even though the average value estimated from simulation might not be quite accurate, it should be safely smaller than the largest DRV value so we are able to find the Vmin point during the upward searching.

Figure 5.4(a) shows the main steps of this upward searching method. The applied standby voltage (VDDS) starts from an initial lower value, VDDS0 (e.g. the value nearest to the average SRAM DRV value from simulation). For each VDDS, the BIST checks hold failures for data '0' and then data '1'. If both the checks complete successfully (i.e. Holdsuccess=1), then the current VDDS is the minimum standby voltage, Vmin; otherwise, the checking process is repeated after increasing VDDS by one step.

Row/column redundancy and ECC are conventionally used for reducing the yield loss due to manufacturing defects and soft errors. For low standby power operation, they can also be used to tolerate data-retention errors so that the minimum standby voltage can be less than the worst DRV in the SRAM [52]. Here, we assume the number of redundant rows

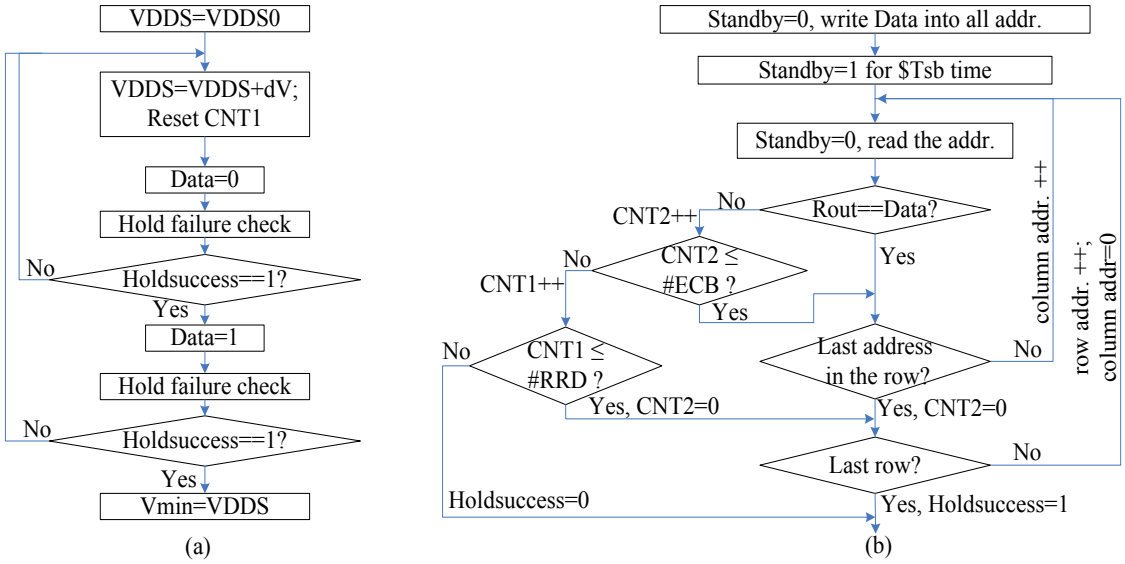


Figure 5.4: (a) Main flow for self-testing SRAM DRV tail in the upward searching manner and (b) Flow for hold failure check.

is #RRD and the number of the correctable bits per row is #ECB. A small register file with #RRD entries saves the row address bits of the replaced row. The detailed flow for checking hold failures in the SRAM with redundancies is illustrated in Figure 5.4(b). First, in active mode, the holding data value is written into each address. Then the SRAM enters standby mode and maintains standby for a period of time ($Tsb). After the standby operation, data is read out and checked in active mode. A counter (CNT2) records the number of failed bits within one row. If it is larger than #ECB, this row needs to be replaced. The BIST then checks CNT1, which stores the number of used redundant rows. If it is less than #RRD, then this row can be replaced and its row address bits will be saved in the register file. Otherwise, the checking process is terminated with Holdsuccess=0. The checking process ends with Holdsuccess=1 only if all of the rows have been checked before exhausting all of the redundant rows.

It should be noted that the duration of the standby operation ($Tsb) should be suffi-

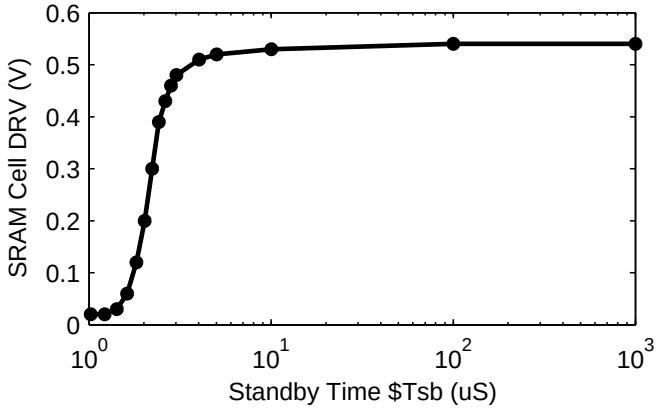


Figure 5.5: The DRV of an SRAM cell changes with the duration of the standby time ($Tsb). With shorter standby time, SRAM cell DRV decreases, i.e. the cell is more stable.

ciently long to ensure the occurrence of the worst static scenario. Figure 5.5 shows an example of one SRAM cell in PTM 22nm. Its DRV first dynamically grows with the increase of the standby time. This is because the cell can tolerate more dynamic noise as logic circuits do [41] when the duration of the noise is shorter. Then after the standby time exceeds a critical point, its DRV reaches the worst value, which actually equals to the value from the static DC simulation.

Figure 5.6 shows the reduction of the test time by using the upward searching method relative to the downward searching method when the Vmin of a 256K-b SRAM varies. Note that for both upward and downward methods, the time consumed for one iteration is mainly determined by the duration of the standby operation for a small or medium size memory. While for a large memory with millions of cells, the time of each iteration is also dependent on the speed of the read and write operation. A reduction of up to $12\times$ can be achieved with the proposed upward searching method. The downward searching method is preferable only when the Vmin is much closer to the nominal V_{DD} ($\geq$0.75) because fewer iterations are needed.

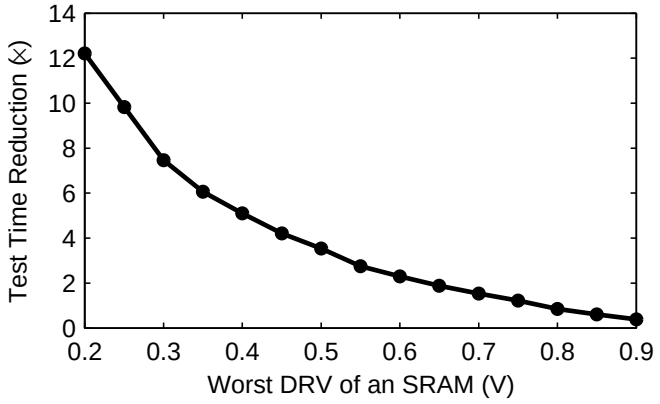


Figure 5.6: Test time reduction by using upward searching method relative to the downward searching method against the worst DRV of a 256K-b SRAM.

5.2.2 Calibrating Initial Failure Threshold

Simulation and measured results have shown that the DRV of the canary cell is approximately linear with VCTRL value [63]. Hence, we can use a series of VCTRL values to create a group of canary categories that fail at regular intervals across a wide range. A resistor ladder with n distinct voltage nodes can generate the series of VCTRL values, which connect to the corresponding canaries. The top of the resistor ladder connects to a reference voltage VREF. So the VCTRL value of the *i*th canary is

$$\text{VCTRL}_i = \frac{i}{n} \cdot \text{VREF}. \tag{5.1}$$

During self-calibration, our BIST first finds the VDDS closest to the actual SRAM DRV tail (Vmin), as discussed in Section 5.2.1. Then the BIST applies VDDS as the supply voltage for canary circuit and measures the failure status of each canary category, FT_{max}. Suppose we get

$$\begin{aligned} FT_{max} &= [f_0 f_1 \cdots f_{k-1} f_k f_{k+1} \cdots f_{n-2} f_{n-1}] \\ &= [00 \cdots 011 \cdots 11]. \end{aligned} \tag{5.2}$$

Here, f_i means the failure status of the ith canary; when f_i=1, this canary fails. So the kth canary is the one that fails immediately before the worst SRAM cell. This FT_{max} value will be recorded (e.g., with a programmable fuse or other non-volatile memory). In normal operation mode, the user first loads FT_{max}, and then programs an appropriate failure threshold value according to the application needs. We denote $FT_{max} >> j$ as the value after right shifting FT_{max} by j bits. For aggressive power saving, the failure threshold register should be configured as $FT_{max} >> 1$; while for more robust V_{DD} scaling, $FT_{max} >> j$ with $j > 1$ should be used to tradeoff less power saving with higher data reliability. It should be noted that the granularity of the tunability of our canary system is dependent on the grid of the VCTRL values and the resolution of the voltage regulator.

5.3 45nm Test Chip Implementation & Measurement

To verify the effectiveness of our scheme in sub-45nm technologies, we implemented the canary circuits in a bulk 45nm test chip. A die photo is shown in Figure 5.7. We put two canary blocks on the die. Each block has 8 canary sets and employs 3-way redundancy for each set. The only difference between the two blocks is the canary cell structure. The first block uses the original canary cell without the dummy cells. On the contrary, the second one uses the improved canary cell with the dummy cells as shown in Figure 5.1. Figure 5.8 shows that the measured canary cell DRV from the 45nm test chip keeps excellent linearity with VCTRL values except for the smaller VCTRL values, which is consistent with our previous measurement result in 90nm. The non-linearity in the range of VCTRL less than 100mV is due to the rolling off term in the sub-threshold current equation [62]. The slope of the curve can be approximated to $1/(1+\eta)$, where η is the DIBL coefficient of the header

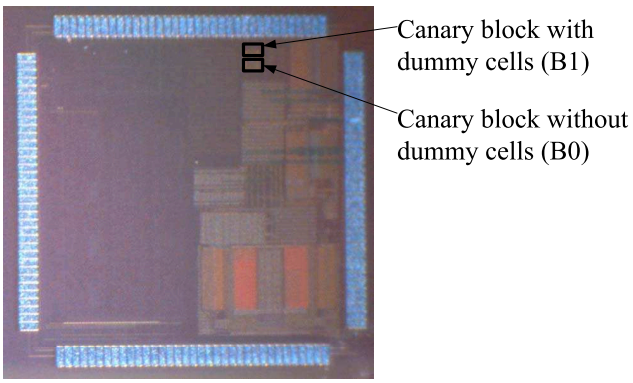


Figure 5.7: 45nm test chip die photo.

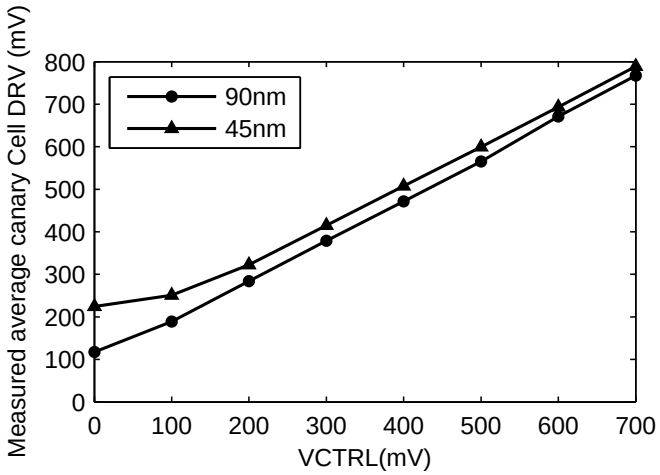
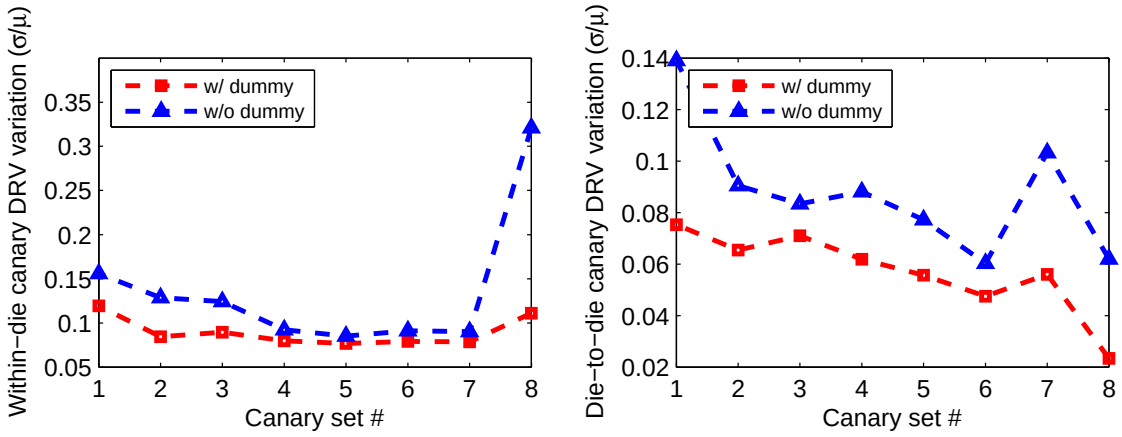


Figure 5.8: Measured average canary DRV vs. VCTRL from this 45nm chip and the previous 90nm chip.

PMOS [62]. Because η increases with technology scaling, the slope of the 45nm curve is slightly smaller than that of the 90nm curve. This implies that the 45nm design needs a larger adjustment of the VCTRL value for the same amount of canary DRV change as for a 90nm design.

Figure 5.9 shows the comparison results between the canary block with and without dummy cells. 85 dies on one wafer are measured. For each die, the VCTRL value of each



(a) (b)

Figure 5.9: Comparison between the canary blocks with and without dummy cells: (a) measured within-die canary DRV variation (average of 85 dies) for each canary set that contains 3 redundancies and (b) measured die-2-die canary DRV variation for each canary set after using majority-3 voting. With dummy cells, both with-in-die and die-2-die variations are reduced.

canary set is generated by an on-die resistor ladder. The canary set with the higher index number connects to a higher VCTRL value. The variation of the canary DRV is computed as the ratio of the sigma (σ) to the mean (μ). A smaller ratio value means less variation occurred on the canary. We first compared the within-die variation, i.e. the variation of the 3 redundancies for each canary set on each die. The average result from 85 dies is plotted in Figure 5.9(a). It shows that the block with the dummy cells has fewer within-die variations, especially for the canary set #8 that is configured to have the largest DRV. We also compared the die-to-die variation. In this case, the canary DRV value of each die is obtained through the majority-3 voting among the redundancies on the same die. Figure 5.9(b) shows that the block with dummy cells also has less die-to-die variations. Therefore, the use of dummy cells inside the canary cell can effectively reduce both with-in-die and die-to-die variations of the canary cell.

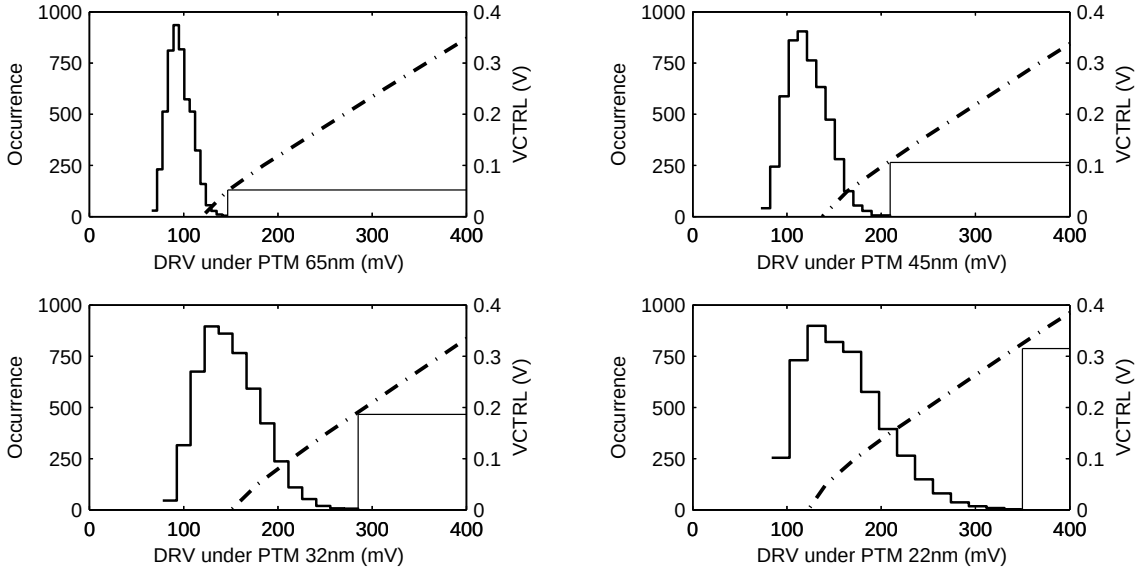


Figure 5.10: 5-Kb SRAM DRV distribution (left axis) and canary DRV vs. VCTRL (right axis) under PTM 65∼32nm nodes.

5.4 Scaling Beyond 45nm

Using Predictive Technology Models (PTMs) for 65nm, 45nm, 32nm and 22nm nodes [10] [43], we investigated the scaling of the canary DRV as well as the SRAM DRV for more advanced technologies. The SRAM cell transistor has the length of Lmin and a width of 2*Lmin (Lmin is the minimum length for the technology). The canary cell header transistor has the same sizing as the other 6T transistors. 3σ of the V_T local variation for 65nm, 45nm, 32nm and 22nm is 10%, 15%, 20% and 25% of the normial V_T, repectively.

Figure 5.10 shows the SRAM DRV distribution and the canary DRV vs. VCTRL for PTM nodes from 65nm to 22nm. The canary cell can keep the linearity property with VCTRL changes for all of the smaller technologies. This means that we can still create a continuum of failure voltages above the actual failure point of the SRAM array down to 22nm. The plots also show that the SRAM DRV distribution has a higher mean and larger

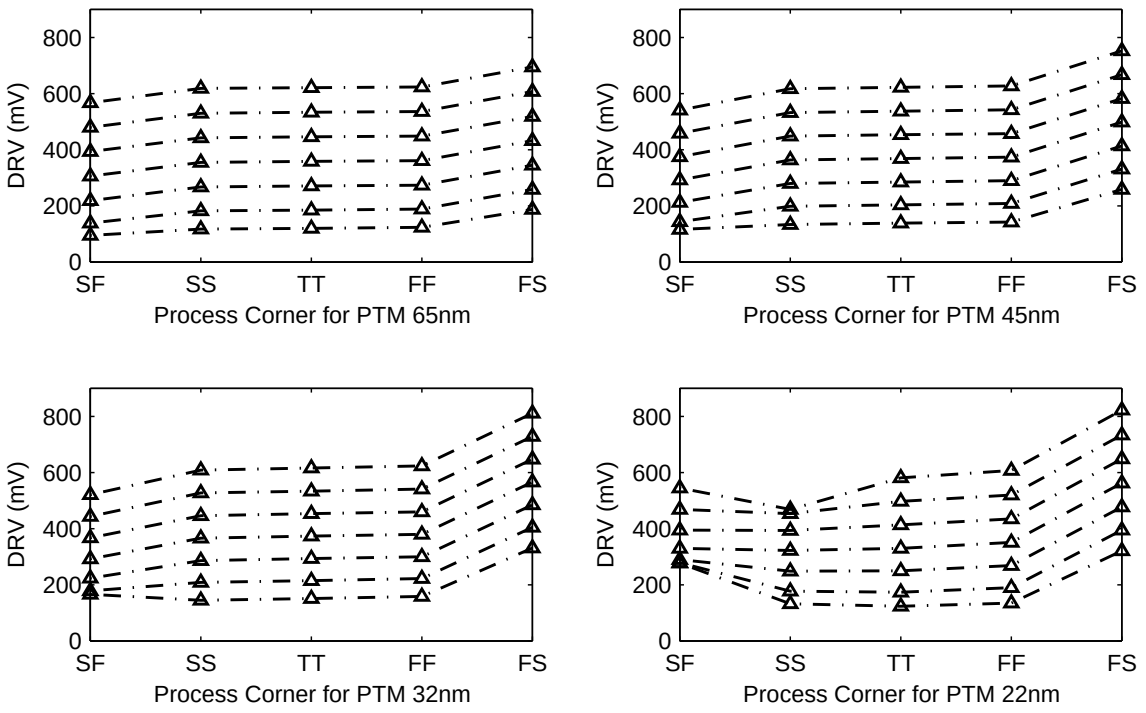
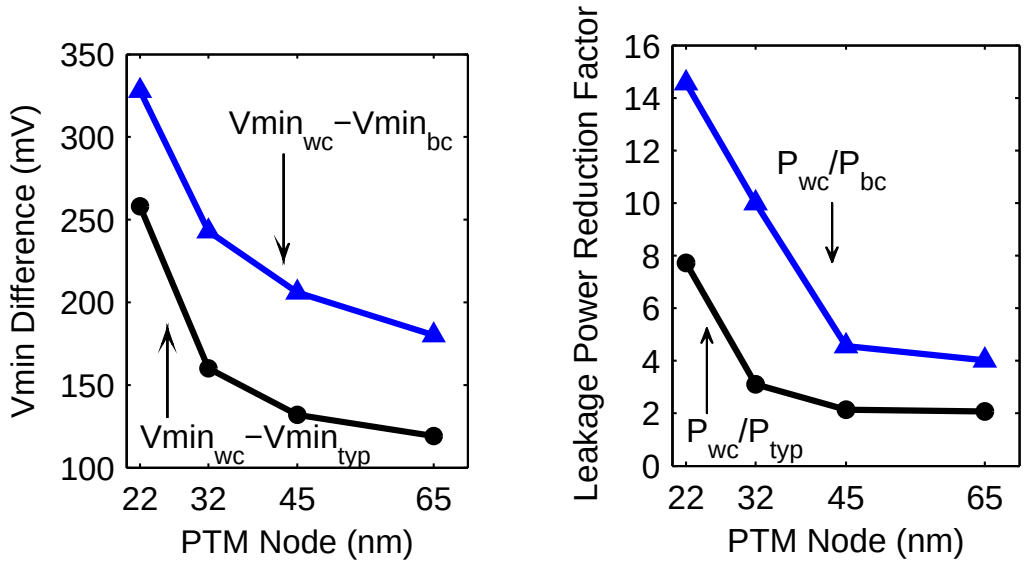


Figure 5.11: DRV of canary categories (each line denotes one category and the upper ones have higher VCTRL values) at different process corners under PTM 65∼32nm nodes.

tail value as well as a larger standard deviation because of the increased variation at smaller dimensions. Therefore, we will need to use higher VCTRL values when using canary cells for SRAMs in smaller technologies to create failures above the DRV of the array.

Figure 5.11 shows that the canary cells can track global process variation for 65nm, 45nm and 32nm nodes. For the 22nm node, because of gate leakage, the canary DRV is no longer linear with the VCTRL (header gate voltage) value at some global process corners when VCTRL is high. This could potentially limit the range over which we can trade off power savings with reliability, but there is enough linearity to successfully deploy the canary scheme at 22nm. If new techniques such as High-K materials provide the anticipated reduction of gate leakage, then the canary scheme will be able to offer a broad range of voltages for this tradeoff. These simulations indicated that our canary scheme can provide



(a) (b)

Figure 5.12: (a) Gap between the Vmin of the worst-case PVT variation ($Vmin_{wc}$) and the Vmin of the best-case/typical-case PVT variation ($Vmin_{bc}$, $Vmin_{typ}$) and (b) Leakage power reduction for using the optimum Vmin (P_{bc} and P_{typ} at the best-case and typical PVT scenario respectively) relative to the leakage power when using the worst case Vmin (P_{wc}). A 1-Kb SRAM is simulated across the PTM bulk technologies from 65nm to 22nm.

effective power reduction for future nodes down to 22nm.

Figure 5.12(a) shows the gap between the Vmin of the worst-case PVT variation and the Vmin of the best-case/typical variation for a 1-Kb SRAM array using PTM models from 65nm to 22nm. The Vmin gap between the worst-case and non-worst-cases grows with technology scaling. Compared with using the worst Vmin, up to 4× and 14× leakage power reduction can be achieved by using the optimum Vmin at 65nm and 22nm respectively as shown in Figure 5.12(b). The worst-case based approach loses more power savings as the variability increases with technology scaling.

Chapter 6

Conclusion

The growing demand of power reduction in SoC systems requires SRAM to operate at a lower voltage. However, lowered voltage degrades SRAM functional yield. To maintain a lowest acceptable yield, SRAM must operate above the minimum operational voltage (Vmin). In this dissertation, we investigate the impact of both local and global variations on SRAM minimum operational voltage, and present two solutions to address each. Local random variations spread the cell Vmin across the same array, and the tail of the distribution determines SRAM Vmin. We propose a fast and accurate method to predict the Vmin value for large SRAMs based on the sensitivity of SNM to V_{DD}. The method is generalized for estimating Vmin due to hold, read, and write failures. Our method offers the comparable accuracy with standard Monte Carlo (MC) and shows excellent agreements with other fast methods, including the Statistical Blockade (SB) tool and Importance Sampling (IS), for the tails up to 8σ. It offers the speed up of $> 10^4$ over MC and shows less complexity and variance than SB and IS. Global PVT variation primarily results in the shift of Vmin values. The worst-case design approach over-protects the non-worst scenarios and limits

power reduction. We propose a closed-loop V_{DD} scaling system that can truly eliminate margins for PVT variation and achieve leakage power reduction near the optimum. It uses online canary replica cells and monitors to track global variations, and a feedback circuit to adjust V_{DD} to approach the true SRAM Vmin. Several techniques are proposed to enhance the adaptiveness of the canary system for SRAMs beyond 45nm. Silicon results from 90nm and 45nm test chip confirm the function of the system.

There are several additional research directions that extend our statistical method for Vmin estimation and the canary based adaptive system.

Extended work for Vmin Estimation

1. Various read and write assist techniques have been proposed to expand 6T SRAM operational margins. The sensitivity of static noise margin to V_{DD} might change with different assist techniques or even different bias value with the same assist knob. Vmin model can be extended for evaluating the effectiveness of assist methods for Vmin/yield improvement.

2. Our current work uses static noise margin (SNM) as the criterion for read/write failures. SNM represents the maximum tolerable dc noise, which is often smaller than dynamic noise margin (DNM). Since the real read/write operation is performed in a dynamic fashion, i.e. with the timing constraint, DNM will be more accurate. The connection of DNM with V_{DD} under variation can improve the accuracy of Vmin and yield estimation.

3. Besides hold, read stability, and write failures, the final Vmin is also determined by read access failure due to an insufficient sensing signal developed within the required

access time. A read access failure is dependent on various factors, including the cell current, the bitline leakage current, the sense amplifier enable time, and the sense amplifier offset. A correct estimation of Vmin/yield due to read access failure must combine these statistical distributions together.

4. Vmin also drifts with aging effects such as NBTI. To estimate Vmin/yield degradation under NBTI, the impact of NBTI on SRAM stability should be investigated.

Extended work for Canary-based Adaptive System

1. The architecture of the canary feedback system can be used for active V_{DD} scaling or DVS system to achieve aggressive active power reduction. Novel canary replica cells for tracking the impact of PVT variation on SRAM read stability failure, write failure, and access failure can be explored.

2. We can develop canary replica cells to prevent failures due to aging issues such as NBTI.

3. A complete closed-loop V_{DD} scaling system needs a on-die voltage regulator, which should operate with a high efficiency in a wide voltage range.

Bibliography

[1] A. Agarwal, Hai Li, and K. Roy. A single-vt low-leakage gated-ground cache for deep submicron. *IEEE J. Solid-State Circuits*, 38(2):319–328, 2003.

[2] A. Asenov, A. R. Brown, J. H. Davies, S. Kaya, and G. Slavcheva. Simulation of intrinsic parameter fluctuations in decananometer and nanometer-scale MOSFETs. *IEEE Trans. Electron Devices*, 50(9):1837–1852, 2003.

[3] A. Bhavnagarwala, S. Kosonocky, Yuen Chan, K. Stawiasz, U. Srinivasan, S. Kowalczyk, and M. Ziegler. A sub-600mv, fluctuation tolerant 65nm CMOS SRAM array with dynamic cell biasing. In *Symp. VLSI Circuits Dig.*, pages 78–79, 2007.

[4] A. Bhavnagarwala, S. Kosonocky, C. Radens, K. Stawiasz, R. Mann, Qiuyi Ye, and K. Chin. Fluctuation limits & scaling opportunities for CMOS SRAM cells. In *IEEE Int. Electron Devices Meeting (IEDM)*, pages 659–662, 5-7 Dec. 2005.

[5] A. J. Bhavnagarwala, S. V. Kosonocky, S. P. Kowalczyk, R. V. Joshi, Y. H. Chan, U. Srinivasan, and J. K. Wadhwa. A transregional CMOS SRAM with single, logic V_{DD} and dynamic power rails. In *Symp. VLSI Circuits Dig.*, pages 292–293, 17–19 June 2004.

[6] D. Blaauw, S. Kalaiselvan, K. Lai, Wei-Hsiang Ma, S. Pant, C. Tokunaga, S. Das, and D. Bull. Razor II: In situ error detection and correction for pvt and ser tolerance. In *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, pages 400–622, 2008.

[7] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De. Parameter

variations and impact on circuits and microarchitecture. In *Proc. Design Automation Conference*, pages 338–342, 2003.

[8] B.H. Calhoun and A. Chandrakasan. A 256kb sub-threshold SRAM in 65nm CMOS. In *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, pages 2592–2601, Feb. 6-9, 2006.

[9] B.H. Calhoun and A.P. Chandrakasan. Standby power reduction using dynamic voltage scaling and canary flip-flop structures. *IEEE J. Solid-State Circuits*, 39(9):1504–1511, Sept. 2004.

[10] Y. Cao, T. Sato, M. Orshansky, D. Sylvester, and C. Hu. New paradigm of predictive MOSFET and interconnect modeling for early circuit simulation. In *Proc. Custom Integrated Circuits Conf. (CICC)*, pages 201–204, 2000.

[11] I. Chang, J. Kim, S. Park, and K. Roy. A 32kb 10T subthreshold SRAM array with bit-interleaving and differential read scheme in 90nm CMOS. In *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2008.

[12] L. Chang, L. Chang, D.M. Fried, J. Hergenrother, J.W. Sleight, R.H. Dennard, R.K. Montoye, L. Sekaric, S.J. McNab, A.W. Topol, C.D. Adams, C.D. A10 Adams, K.W. Guarini, K.W. A11 Guarini, and W. Haensch, W. A12 Haensch. Stable SRAM cell design for the 32 nm node and beyond. In *Symp. VLSI Technology Dig.*, pages 128–129, 2005.

[13] L. Chang, Y. Nakamura, R.K. Montoye, J. Sawada, A.K. Martin, K. Kinoshita, F.H. Gebara, K.B. Agarwal, D.J. Acharyya, W. Haensch, K. Hosokawa, and D. Jamsek. A 5.3ghz 8t-SRAM with operation down to 0.41v in 65nm CMOS. In *Symp. VLSI Circuits Dig.*, pages 252–253, 14–16 June 2007.

[14] Y. H. Chen, W. M. Chan, S. Y. Chou, H. J. Liao, H. Y. Pan, J. J. Wu, C. H. Lee, S. M. Yang, Y. C. Liu, and H. Yamauchi. A 0.6v 45nm adaptive dual-rail SRAM compiler circuit design for lower vddmin vlsis. In *Symp. VLSI Circuits Dig.*, pages 210–211, 18–20 June 2008.

[15] L. Dolecek, M. Qazi, D. Shah, and A. P. Chandrakasan. Breaking the simulation barrier: SRAM evaluation through norm minimization. In *Proc. IEEE/ACM Int. Conf. on Computer-Aided Design ICCAD 2008*, pages 322–329, 2008.

[16] T. S. Doorn, E. J. W. ter Maten, J. A. Croon, A. Di Bucchianico, and O. Wittich. Importance sampling monte carlo simulations for accurate estimation of SRAM yield. In *Proc. 34th European Solid-State Circuits Conf. (ESSCIRC)*, pages 230–233, 15–19 Sept. 2008.

[17] A. Drake, R. Senger, H. Deogun, G. Carpenter, S. Ghiasi, T. Nguyen, N. James, M. Floyd, and V. Pokala. A distributed critical-path timing monitor for a 65nm high-performance microprocessor. In *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, pages 398–399, February 11–15, 2007.

[18] T. Enomoto, Y. Oka, and H. Shikano. A self-controllable voltage level (svl) circuit and its low-power high-speed CMOS circuit applications. *IEEE J. Solid-State Circuits*, 38(7):1220–1226, 2003.

[19] A. Ferre and J. Figueras. Leakage in cmos nanometric technologies. In C. Piguet, editor, *Low-Power Electronics Design*, pages 3-6. CRC Press, Boca Raton, second edition, 2005.

[20] K. Flautner, Nam Sung Kim, S. Martin, D. Blaauw, and T. Mudge. Drowsy caches: simple techniques for reducing leakage power. In *Proc. Int. Symp. Computer Architecture*, pages 148–157, 25-29 May 2002.

[21] S. Ghosh, S. Mukhopadhyay, Keejong Kim, and K. Roy. Self-calibration technique for reduction of hold failures in low-power nano-scaled SRAM. In *Proc. Design Automation Conf. (DAC)*, pages 971–976, 24-28 July 2006.

[22] N. Gierczynski, N. Gierczynski, B. Borot, N. Planes, and H. Brut. A new combined methodology for write-margin extraction of advanced SRAM. In *Proc. IEEE Int. Conf. on Microelectronic Test Structures (ICMTS)*, pages 97–100, 2007.

[23] E. Grossar, M. Stucchi, K. Maex, and W. Dehaene. Read stability and write-ability analysis of SRAM cells for nanometer technologies. *IEEE J. Solid-State Circuits*, 41(11):2577–2588, Nov. 2006.

[24] C. Gu and J. Roychowdhury. An efficient, fully nonlinear, variability-aware non-monte-carlo yield estimation procedure with applications to SRAM cells and ring oscillators. In *Proc. Asia and South Pacific Design Automation Conf. (ASPDAC'08)*, pages 754–761, 21–24 March 2008.

[25] F. Hamzaoglu, Kevin Zhang, et al. A 153mb-SRAM design with dynamic stability enhancement and leakage reduction in 45nm high-k metal-gate CMOS technology. In *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, pages 376–621, 3–7 Feb. 2008.

[26] R. Heald and P. Wang. Variability in sub-100nm SRAM designs. In *Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM Int. Conf. on*, pages 347–352, 7-11 Nov. 2004.

[27] T. C. Hesterberg. *Adavances in Importance Sampling*. PhD thesis, Stanford University, 1988.

[28] R. Joshi, R. Houle, K. Batson, D. Rodko, P. Patel, W. Huott, R. Franch, Y. Chan, D. Plass, S. Wilson, S. A10 Wilson, and P. Wang, P. A11 Wang. 6.6+ ghz low vmin, read and half select disturb-free 1.2 mb SRAM. In *Symp. VLSI Circuits Dig.*, pages 250–251, 2007.

[29] K. Kanda, T. Miyazaki, Min Kyeong Sik, H. Kawaguchi, and T. Sakurai. Two orders of magnitude leakage power reduction of low voltage SRAMs by row-by-row dynamic Vdd control (RRDV) scheme. In *Proc. IEEE Int. ASIC/SOC Conference*, pages 381–385, Sept. 2002.

[30] R. Kanj, R. Joshi, and S. Nassif. Mixture importance sampling and its application to the analysis of SRAM designs in the presence of rare failure events. In *Proc. 43rd ACM/IEEE Design Automation Conf.*, pages 69–72, 2006.

[31] M. Khellah, N. S. Kim, Y. Ye, et al. Pvt-variations and supply-noise tolerant 45nm dense cache arrays with diffusion-notch-free (dnf) 6t SRAM cells and dynamic multi-vcc circuits. In *Symp. VLSI Circuits Dig.*, pages 48–49, 2008.

[32] M. Khellah, D. Somasekhar, Y. Ye, et al. A 256-kb dual-vcc SRAM building block in 65-nm CMOS process with actively clamped sleep transistor. *IEEE J. Solid-State Circuits*, 42(1):233–242, 2007.

[33] M. Khellah, Yibin Ye, Nam Sung Kim, D. Somasekhar, G. Pandya, A. Farhang, K. Zhang, C. Webb, and V. De. Wordline &bitline pulsing schemes for improving SRAM cell stability in low-Vcc 65nm CMOS designs. In *Symp. VLSI Circuits Dig.*, pages 9–10, 2006.

[34] C.H. Kim, Jae-Joon Kim, S. Mukhopadhyay, and K. Roy. A forward body-biased low-leakage SRAM cache: device, circuit and architecture considerations. *IEEE Trans. VLSI Syst.*, 13(3):349–357, March 2005.

[35] Nam Sung Kim, K. Flautner, D. Blaauw, and T. Mudge. Single-vdd and single-vt super-drowsy techniques for low-leakage high-performance instruction caches. In *Int. Symp. Low Power Electronics and Design (ISLPED)*, pages 54–57, 2004.

[36] T-H Kim, J. Liu, J. Keane, and C.H. Kim. A high-density subthreshold SRAM with data-independent bitline leakage and virtual ground replica scheme. In *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, pages 330–606, 11–15 Feb. 2007.

[37] R.K. Krishnamurthy, A. Alvandpour, G. Balamurugan, N.R. Shanbhag, K. Soumyanath, and S.Y. Borkar. A 130-nm 6-ghz 256 × 32 bit leakage-tolerant register file. *IEEE J. Solid-State Circuits*, 37(5):624–632, 2002.

[38] J.P. Kulkarni, K. Kim, and K. Roy. A 160 mv robust schmitt trigger based subthreshold SRAM. *IEEE J. Solid-State Circuits*, 42(10):2303–2313, 2007.

[39] G. La Rosa, Wee Loon Ng, S. Rauch, R. Wong, and J. Sudijono. Impact of nbti induced statistical variation to SRAM cell stability. In *Proc. 44th Annual IEEE Int. Reliability Physics Symp.*, pages 274–282, 26–30 March 2006.

[40] W-C Lee and C. Hu. Modeling CMOS tunneling currents through ultrathin gate oxide due to conduction- and valence-band electron and hole tunneling. *IEEE Trans. Electron Devices*, 48(7):1366–1373, Jul 2001.

[41] J. Lohstroh. Static and dynamic noise margins of logic circuits. *IEEE J. Solid-State Circuits*, 14(3):591–598, 1979.

[42] K. Mistry et al. A 45nm logic technology with high-k+metal gate transistors, strained silicon, 9 cu interconnect layers, 193nm dry patterning, and 100packaging. In *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, pages 247–250, 2007.

[43] Predictive Technology Model. `http://www.eas.asu.edu/~ptm`.

[44] Y. Morita, H. Fujiwara, H. Noguchi, K. Kawakami, J. Miyakoshi, S. Mikami, K. Nii, H. Kawaguchi, and M. Yoshimoto. A vth-variation-tolerant SRAM with 0.3-v minimum operation voltage for memory-rich soc under dvs environment. In *Symp. VLSI Circuits Dig.*, pages 13–14, 2006.

[45] S. Mukhopadhyay, K. Kim, H. Mahmoodi, and K. Roy. Design of a process variation tolerant self-repairing SRAM for yield enhancement in nanoscaled CMOS. *IEEE J. Solid-State Circuits*, 42(6):1370–1382, 2007.

[46] S. Mukhopadhyay, Keejong Kim, H. Mahmoodi, A. Datta, Dongkyu Park, and K. Roy. Self-repairing SRAM for reducing parametric failures in nanoscaled memory. In *Symp. VLSI Circuits Dig.*, pages 132–133, 2006.

[47] Y. Nakagome, M. Horiguchi, T. Kawahara, and K. Itoh. Review and future prospects of low-voltage ram circuits. *IBM Journal of Research & Development*, 47:525–552, 2003.

[48] K. Nii, Y. Tsukamoto, T. Yoshizawa, S. Imaoka, Y. Yamagami, T. Suzuki, A. Shibayama, H. Makino, and S. Iwade. A 90-nm low-power 32-kb embedded SRAM with gate leakage suppression circuit for mobile applications. *IEEE J. Solid-State Circuits*, 39(4):684–693, April 2004.

[49] S. Ohbayashi, M. Yabuuchi, K. Nii, Y. Tsukamoto, S. Imaoka, Y. Oda, T. Yoshihara, M. Igarashi, M. Takeuchi, H. Kawashima, H. A10 Kawashima, Y. Yamaguchi, Y. A11 Yamaguchi, K. Tsukamoto, K. A12 Tsukamoto, M. Inuishi, M. A13 Inuishi, H. Makino, H. A14 Makino, K. Ishibashi, K. A15 Ishibashi, and H. Shinohara, H. A16 Shinohara. A 65-nm SoC embedded 6t-SRAM designed for manufacturability with read and write operation stabilizing circuits. *IEEE J. Solid-State Circuits*, 42(4):820–829, 2007.

[50] H. Pilo, C. Barwin, G. Braceras, C. Browning, S. Lamphier, and F. Towler. An SRAM design in 65-nm technology node featuring read and write-assist circuits to expand operating voltage. *IEEE J. Solid-State Circuits*, 42(4):813–819, 2007.

[51] H. Qin, Y. Cao, D. Markovic, A. Vladimirescu, and J. Rabaey. SRAM leakage suppression by minimizing standby supply voltage. In *Proc. Int. Symp. Quality Electronic Design (ISQED)*, pages 55–60, 2004.

[52] H. Qin, A. Kumar, K. Ramchandran, J. Rabaey, and P. Ishwar. Error-tolerant SRAM design for ultra-low power standby operation. In *Proc. Int. Symp. on Quality Electronic Design (ISQED)*, pages 30–34, 2008.

[53] Y.K. Ramadass and A.P. Chandrakasan. Minimum energy tracking loop with embedded dc-dc converter delivering voltages down to 250mv in 65nm CMOS. In *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, pages 64–587, 2007.

[54] W. M. Randy, J. Wang, S Nalam, S Khanna, G. Braceras, H Pilo, and B. H. Calhoun. Methodology for evaluation of 6t SRAM assist methods in scaled CMOS technologies. *submitted to IEEE J. Solid-State Circuits*.

[55] E. Seevinck, F.J. List, and J. Lohstroh. Static-noise margin analysis of mos SRAM cells. *IEEE J. Solid-State Circuits*, 22(5):748–754, 1987.

[56] N. Shibata, H. Kiya, S. Kurita, H. Okamoto, M. Tan'no, and T. Douseki. A 0.5-v 25-mhz 1-mw 256-kb mtCMOS/soi SRAM for solar-power-operated portable personal digital equipment - sure write operation by using step-down negatively overdriven bitline scheme. *IEEE J. Solid-State Circuits*, 41(3):728–742, March 2006.

[57] A. Singhee and R.A. Rutenbar. Statistical blockade: A novel method for very fast monte carlo simulation of rare circuit events, and its application. In *Proc. Design, Automation & Test in Europe Conf. & Exhibition DATE '07*, pages 1–6, 2007.

[58] S. Srivastava and J. Roychowdhury. Rapid estimation of the probability of SRAM failure due to mos threshold variations. In *Proc. IEEE Custom Integrated Circuits Conf. CICC '07*, pages 229–232, 2007.

[59] Y. Takeyama, H. Otake, O. Hirabayashi, K. Kushida, and N. Otsuka. A low leakage SRAM macro with replica cell biasing scheme. *IEEE J. Solid-State Circuits*, 41(4):815–822, 2006.

[60] J.W. Tschanz, J.T. Kao, S.G. Narendra, R. Nair, D.A. Antoniadis, A.P. Chandrakasan, and V. De. Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage. *IEEE J. Solid-State Circuits*, 37(11):1396–1402, 2002.

[61] N. Verma and A.P. Chandrakasan. A 256 kb 65 nm 8t subthreshold SRAM employing sense-amplifier redundancy. *IEEE J. Solid-State Circuits*, 43(1):141–149, 2008.

[62] J. Wang and B. H. Calhoun. Techniques to extend canary-based standby v_{DD} scaling for SRAMs to 45 nm and beyond. *IEEE J. Solid-State Circuits*, 43(11):2514–2523, Nov. 2008.

[63] J. Wang and B.H. Calhoun. Canary replica feedback for near-drv standby V_{DD} scaling in a 90nm SRAM. In *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, pages 29–32, 2007.

[64] J. Wang, S. Nalam, and B. H. Calhoun. Analyzing static and dynamic write margin for nanometer SRAMs. In *Proc. Int. Symp. on Low power electronics and design*, pages 129–134, New York, NY, USA, 2008. ACM.

[65] J. Wang, A. Singhee, R. A. Rutenbar, and B. H. Calhoun. Statistical modeling for the minimum standby supply voltage of a full SRAM array. In *Proc. European Solid State Circuits Conf. (ESSCIRC)*, pages 400–403, 2007.

[66] T. Wang, T.E. Chang, C.M. Huang, J.Y. Yang, K.M. Chang, and L.P. Chiang. Structural effect on band-trap-band tunneling induced drain leakage in n-MOSFET's. *IEEE Trans. Electron Devices*, 16(12):566–568, Dec 1995.

[67] Y. Wang, U. Bhattacharya, F. Hamzaoglu, P. Kolar, Y. Ng, L. Wei, Y. Zhang, K. Zhang, and M. Bohr. A 4.0 GHz 291Mb voltage-scalable SRAM design in 32nm high-k metal-gate CMOS with integrated power management. In *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, pages 456–457,457a, February 8–12, 2009.

[68] Y. Wang et al. A 1.1ghz 12μA/Mb-leakage SRAM design in 65nm ultra-low-power CMOS with integrated leakage reduction for mobile applications. In *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, pages 324–606, 2007.

[69] C. Wann, R. Wong, D.J. Frank, R. Mann, Shang-Bin Ko, P. Croce, D. Lea, D. Hoyniak, Yoo-Mi Lee, J. Toomey, M. Weybright, and J. Sudijono. SRAM cell design for stability methodology. In *IEEE Int. Symp. VLSI Technology (VLSI-TSA-Tech)*, pages 21–22, April 2005.

[70] J. Watts, N. Lu, C. Bittner, S. Grundon, and J. Oppold. Modeling fet variation within a chip as a function of circuit design and layout choices. In *Nanotech Workshop on Compact Modeling*, pages 87–92, 2005.

[71] M. Yamaoka. A 65nm low-power high-density SRAM operable at 1.0v under 3σ systematic variation using separate vth monitoring and body bias for NMOS and PMOS. In *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, pages 384–622, 2008.

[72] M. Yamaoka, Y. Shinozaki, N. Maeda, Y. Shimazaki, K. Kato, S. Shimada, K. Yanagisawa, and K. Osada. A 300-MHz 25-μA/Mb-leakage on-chip SRAM module featuring process-variation immunity and low-leakage-active mode for mobile-phone application processor. *Solid-State Circuits, IEEE Journal of*, 40(1):186–194, 2005.

[73] Y. Ye, M. Khellah, D. Somasekhar, A. Farhang, and V. De. A 6-ghz 16-kb l1 cache in a 100-nm dual-v/sub t/ technology using a bitline leakage reduction (blr) technique. *IEEE J. Solid-State Circuits*, 38(5):839–842, 2003.

[74] H-S Yu, N-S Kim, Y-J Son, Y-G Kim, H-C Kim, U-R Cho, and H-G Byun. A SRAM core architecture with adaptive cell bias scheme. In *Symp. VLSI Circuits Dig.*, pages 128–129, 2006.

[75] B. Zhai, D. Blaauw, D. Sylvester, and S. Hanson. A sub-200mv 6t SRAM in 0.13um CMOS. In *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, pages 332–606, 11–15 Feb. 2007.

[76] K. Zhang, U. Bhattacharya, Zhanping Chen, F. Hamzaoglu, D. Murray, N. Vallepalli, Yih Wang, B. Zheng, and M. Bohr. SRAM design on 65-nm CMOS technology with dynamic sleep transistor for leakage reduction. *IEEE J. Solid-State Circuits*, 40(4):895–901, 2005.

[77] K. Zhang, U. Bhattacharya, Zhanping Chen, F. Hamzaoglu, D. Murray, N. Vallepalli, Yih Wang, Bo Zheng, and M. Bohr. A 3-ghz 70-mb SRAM in 65-nm CMOS technology with integrated column-based dynamic power supply. *IEEE J. Solid-State Circuits*, 41(1):146–151, Jan. 2006.